NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

REAL-TIME DYNAMIC MODEL LEARNING AND ADAPTATION FOR UNDERWATER VEHICLES

by

Joshua D. Weiss

September 2013

Thesis Advisor: Noel du Toit
Second Reader: Douglas Horner

Approve for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** September 2013	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE REAL-TIME DYNAMIC MODEL LEARNING AND ADAPTATION FOR UNDERWATER VEHICLES		**5. FUNDING NUMBERS**	
6. AUTHOR(S) Joshua D. Weiss			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000		**8. PERFORMING ORGANIZATION REPORT NUMBER**	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A		**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. government. IRB protocol number ____N/A____.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited		**12b. DISTRIBUTION CODE** A	

13. ABSTRACT (maximum 200 words)

Precision control of unmanned underwater vehicles (UUVs) requires accurate knowledge of the dynamic characteristics of the vehicles. However, developing such models are time and resource intensive. The problem is further exacerbated by the sensitivity of the dynamic model to vehicle configuration. This is particularly true for hovering-class UUVs since sensor payloads are often mounted outside the vehicle body. Methods are investigated in this thesis to learn the dynamic model for such a hovering-class UUV in real time from motion and position measurements. Several system identification techniques, including gradient estimation, Bayesian estimation, neural network estimation, and recursive linear least square estimation, are employed to estimate equations of motion coefficients. Experimental values are obtained for the surge, sway, heave, and yaw degrees of freedom. Theoretical results are obtained for the roll and pitch degrees of freedom. The experimentally obtained model is then compared to the true vehicle behavior.

14. SUBJECT TERMS Unmanned underwater vehicles; system identification; hydrodynamic model; online model learning; autonomous underwater system			**15. NUMBER OF PAGES** 107
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

REAL-TIME DYNAMIC MODEL LEARNING AND ADAPTATION FOR UNDERWATER VEHICLES

Joshua D. Weiss
Lieutenant, United States Navy
B.S., United States Naval Academy, 2005

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 2013

Author: Joshua D. Weiss

Approved by: Noel du Toit
Thesis Advisor

Douglas Horner
Second Reader

Knox T. Millsaps
Chair, Department of Mechanical and Aerospace Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Precision control of unmanned underwater vehicles (UUVs) requires accurate knowledge of the dynamic characteristics of the vehicles. However, developing such models are time and resource intensive. The problem is further exacerbated by the sensitivity of the dynamic model to vehicle configuration. This is particularly true for hovering-class UUVs since sensor payloads are often mounted outside the vehicle body. Methods are investigated in this thesis to learn the dynamic model for such a hovering-class UUV in real time from motion and position measurements. Several system identification techniques, including gradient estimation, Bayesian estimation, neural network estimation, and recursive linear least square estimation, are employed to estimate equations of motion coefficients. Experimental values are obtained for the surge, sway, heave, and yaw degrees of freedom. Theoretical results are obtained for the roll and pitch degrees of freedom. The experimentally obtained model is then compared to the true vehicle behavior.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I. INTRODUCTION....1
A. MOTIVATION....1
B. LITERATURE REVIEW....2
C. SCOPE OF THIS WORK....5

II. SEABOTIX VLBV300 REMOTELY OPERATED VEHICLE....9
A. OVERVIEW....9
B. VEHICLE SPECIFICATIONS....10
C. CAVR VEHICLE CONFIGURATION....11
D. PROPULSION MODEL....12

III. GENERIC MOTION MODEL....17
A. FULL EQUATIONS OF MOTION....17
B. ASSUMPTIONS AND SIMPLIFIED EQUATIONS....22

IV. SYSTEM IDENTIFICATION....25
A. PARAMETER ESTIMATION FOR A STATIC SYSTEM....25
1. Recursive Linear Least Squares....25
2. Neural Network....26
3. Gradient Estimator....32
4. Bayesian Filtering....34
B. PARAMETER ESTIMATION OF A DYNAMIC SYSTEM....38
C. PERSISTENCE OF EXCITATION....39

V. SYSTEM IDENTIFICATION APPLIED TO VLBV300....43
A. RLLS ESTIMATION....45
1. RLLS Estimation Applied to Simulator....45
2. RLLS Applied to SeaBotix vLBV300....47
3. Conclusion: RLLS Estimator....61
B. NEURAL NETWORK ESTIMATION....63
1. Neural Network Applied to Simulator....63
2. Neural Network applied to SeaBotix vLBV300....68
C. GRADIENT ESTIMATOR....71
D. BAYESIAN FILTERING....73

VI. CONCLUSIONS....77
A. SUMMARY....77
B. FUTURE WORK....79

APPENDIX A. 6 DOF MOTION SIMULATOR MATLB CODE....81

APPENDIX B. SIMULINK DIAGRAMS....85

LIST OF REFERENCES....87

INITIAL DISTRIBUTION LIST....89

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

Figure 1. SeaBotix vLBV300 miniROV platform (From [15]) has been modified to allow tethered, autonomous operations....9
Figure 2. Horizontal and vertical thruster configuration of the vLBV300....11
Figure 3. Body reference frame of the vLBV300....12
Figure 4. Thruster testing showing surface effects....15
Figure 5. Bounded propulsion model for a single thruster on the vLBV300 compared with experimental data....16
Figure 6. A basic neural network showing the network output expressed as a sum of weighted inputs....27
Figure 7. Diagram of an adaptive neural network (After [16])....29
Figure 8. Network architecture of a NARX created in Matlab....31
Figure 9. Convergence of simulator surge parameters....46
Figure 10. SeaBotix vLBV300 in the instrumented NPS dive tank....48
Figure 11. Simulated surge displacement compared to experimental surge displacement at a PWM command of 20....50
Figure 12. Simulated surge velocity compared to experimental surge velocity at a PWM command of 20....50
Figure 13. Simulated surge displacement results compared to measured surge displacement results at a PWM command of 50 using coefficients determined at a PWM command of 50....54
Figure 14. Simulated surge displacement results compared to measured surge displacement results at a PWM command of 20 using coefficients determined at a PWM command of 20....54
Figure 15. Comparison of surge displacement to experimental displacement using coefficients determined at PWM command of 50 and a simulated run at a PWM command of 20....55
Figure 16. Surge displacement comparison using linear only damping model....59
Figure 17. Convergence of surge coefficients at high speed....60
Figure 18. Convergence of heave coefficients at high speed....61
Figure 19. Diagnostic plot of NARX with three nodes and two time delays....64
Figure 20. Diagnostic plot of NARX with ten node and two time delays....64
Figure 21. Comparison of actual surge velocities to ten node and three node NARX....65
Figure 22. Diagnostic results for surge element of complex NARX....67
Figure 23. Comparison of surge velocity response....68
Figure 24. Diagnostic results for velocity mapping, three node, two delay NARX in surge direction only....69
Figure 25. Comparison of three and one node NARXs to measured data....70
Figure 26. GE results in the surge direction with no noise added....72
Figure 27. GE results in the surge direction with added noise....73
Figure 28. RLLS estimator....85
Figure 29. Gradient estimator....85
Figure 30. Simulator....86

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 1. General characteristics (From [15]) 10
Table 2. Tether characteristics (From [15]) 11
Table 3. Comparison of hydrodynamic parameters 47
Table 4. Normalization variables used in the Prime I system (From [1]) 51
Table 5. Parameter estimation results for SeaBotix vLBV300 at high speed 61

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AUV	Autonomous Underwater Vehicle
CAVR	Center for Autonomous Vehicle Research
DOF	Degree(s) of Freedom
EBP	Error Back Propagation
EOD	Explosive Ordnance Disposal
GE	Gradient Estimator
HD	High Definition
NARX	Nonlinear Autoregressive (with External Input) Neural Network
NIO	Nonlinear Input-Output Neural Network
NN	Neural Network
PA	Port, aft propeller on the vLBV300
PE	Persistence of Excitation
PF	Port, forward propeller on the vLBV300
PWM	Pulse Width Modulation
RLLS	Recursive Linear Least Square Estimator
ROV	Remotely Operated Vehicle
RPM	Revolutions per Minute
SA	Starboard, aft propeller on the vLBV300
SF	Starboard, forward propeller on the vLBV300
SV	Starboard, vertical propeller on the vLBV300
THAUS	Tethered Hovering-Class Autonomous Underwater System
UUV	Unmanned Underwater Vehicle
VP	Port, Vertical propeller on the vLBV300

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

First and foremost, I would like to thank my beautiful wife, Alisha, for her love and support throughout this process. Without her wit, intelligence, and friendship none of this work would have been possible. I would also like to greatly thank Dr. Noel du Toit, my adviser, for his support and patience. His dedication to the work and his students is unparalleled at the Naval Postgraduate School. Finally, I would like to thank Dr. Vladimir Dobrokhodov for his incredibly valuable assistance and his willingness to drop whatever he is doing to share his vast knowledge with a confused student.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. MOTIVATION

Historically, robotics research has focused on individual capabilities, such as traveling between points and obstacle avoidance. However, a fundamental shift is occurring: robots are increasingly being put to work in real-world environments. These environments tend to be complex and cluttered, and the tasks are complicated, requiring advances in controls, sensing, perception, and communication. In particular, dive operations are inherently dangerous. Physiological effects limit dive duration and frequency and necessitate a large support crew, increasing operational costs. The sensory-deprived underwater environment makes navigation, communication, and documentation challenging. The Center for Autonomous Vehicle Research (CAVR) at the Naval Postgraduate School (NPS) is developing a Robotic Diver Assistant System (RDAS) to provide autonomous support to diver teams, which has the potential to significantly enhance underwater operations. The RDAS project is aimed at providing utility to the diver team (e.g., illumination, improved situational awareness, etc.) without burdening the team with vehicle command and control, thereby augmenting the diver team and allowing more effective, efficient, and safer operations. This program seeks to go beyond co-inhabitance of man and machine—the aim is to fundamentally enable the transformative capability of robots as underwater co-workers.

The RDAS finds application in many naval operations, but of particular interest is the potential benefit to the salvage, explosive ordinance disposal (EOD), and undersea rescue operations of the Department of the Navy. This application requires operation of a hovering-class unmanned underwater vehicle (UUV) in close proximity to humans as well as other features (e.g., structures, the sea bottom, etc.), which in turn requires precision control of the vehicle. The application will require testing of a vehicle in multiple configurations (i.e., different payload combinations) as well as the development of various perception and control strategies. An accurate dynamic model is required to facilitate precision control. However, these hydrodynamic models are notoriously time and resource consuming to develop in practice. Developing individual models for all

possible configurations is impractical. An effective and efficient method for learning the dynamic model *online* is necessary to execute the RDAS program as well as other closed-quarters operations.

The goal of this research is to learn a parameterized dynamic model for an UUV in real time, and to automate the procedure where possible. Each change in payload configuration changes the weight distribution, center of buoyancy, and drag properties of the platform (i.e., changes the dynamics), requiring a new model. Initially, a propulsion model is developed using experimentally obtained thruster data, then the full equations of motion for a free body are developed. Then, several assumptions are made to simplify these equations which are in turn used to create a vehicle simulator. Various techniques in System Identification and probabilistic state estimation (that account for uncertainties in the measurement and modeling process) are investigated to develop a parametric model of the system as well as a non-parametric, neural-network-based approach. These methods are investigated to determine which of the four are able to be used for this application as well as the strong and weak points of these methods. The applicability of, and results from, these results are tested using the developed simulator and then either applied to the vLBV300 or discarded. This work culminates in a hydrodynamic model of the SeaBotix vLBV300 remotely operated vehicle (ROV), which is the development platform for the CAVR robotic diver assistant.

B. LITERATURE REVIEW

An accurate dynamic model for an UUV is required to inform the parameter learning process, for simulation of a vehicle, and for validation of experimental results. Fossen [1] presents a detailed approach exploring the kinematics and dynamics that make up a complete model for a generic six degree of freedom underwater vehicle. He presents an approach using the Newton-Euler formulation based on Newton's second law

$$m\dot{\upsilon}_C = f_C \tag{1}$$

where m is the mass of a body, $\dot{\upsilon}_C$ is acceleration and f_C is force acting on the body. It can easily be seen from Equation (1) that if no force is acting on the body then

acceleration will be zero and the body will move at a constant speed. Fossen then describes Euler's first and second axioms to Newton's second law,

$$\dot{p}_C \triangleq f_C \qquad p_C \triangleq m\upsilon_C \tag{2}$$

$$\dot{h}_C \triangleq m_C \qquad h_C \triangleq I_C\omega \tag{3}$$

where f_C and m_C are the forces and moments acting on a body referenced to its center of gravity, I_C is the inertia tensor about the body's center of gravity, and ω is the angular velocity vector. These axioms describe Equation (1) in terms of the conservation of linear, p_C, and angular momentum, h_C, which is very convenient. Fossen applies these concepts to develop his generic, six degree of freedom (DOF), nonlinear, and dynamic equations of motion. This generic model is then specialized for the development platform to account for the unique design, relatively slow speeds of operation, and high level of maneuverability. The results from [1] are extensively used in this research.

System identification, also known as model learning, attempts to learn model parameters by systematically exciting various dynamic modes of the vehicle and is an established research field. System identification techniques are distinguished by when they are applied to gathered data (online vs. offline) and what type of model is of interest (parametric vs. non-parametric).

Offline approaches capture the vehicle motion for a commanded behavior and then choose parameters that best represent all the data offline (i.e., regression techniques, such as Linear Least Squares Regression, which minimize the compounded error for the whole data set). Slotine and Li [2] and Astrom and Wittenmark [8] present various analytical and computational methods, including the gradient estimator (which minimizes the instantaneous estimation error) and versions of the least square estimator for system identification to handle time-varying parameters and ill-conditioned learning problems.

Alternatively, model parameters can be estimated online: the estimate is continually updated as new data becomes available. The estimation process itself is considered as a dynamic process, to which stability and convergence analyses can be applied. It is often possible to adapt offline techniques for online implementation (e.g.,

recursive least squares regression), or to develop new techniques entirely (e.g., Gradient Descent methods) [2, 8]. Andrieu *et al.* [12] investigates a method for online, point estimation of static parameters for general state-space models. Kugler [13] presents a non-linear parameter to output operator approach that allows it to be applied to both finite and infinite dimensional differential equations. Thrun and Bulgard [3] present a probabilistic approach based on the Bayes filtering process for online state estimation that accounts for noise and uncertainty in the state and process measurements. The Bayesian approach is premised on the concepts of controllability and observability. A method for determining both the controllability and observability of a system is presented by Ogata [18].

Offline system identification techniques have been applied to UUVs. Doherty [6] and Prestero [7] both employed various analytical techniques based on assumptions, geometry, and empirical formulae to determine the hydrodynamic coefficients of the Hydroid REMUS100 UUV. Prestero's work dealt with a typical REMUS while Doherty derived coefficients for the "long-body" REMUS equipped with cross body thrusters. Both authors used experimental data to confirm their analytical findings. Chen *et al.* [5] used a projective mapping method to estimate the dynamic parameters of a hovering class UUV similar to the work presented here. A vision-based system is used to capture several images of a UUV during specialized maneuvers to track the vehicle position. Eng *et al.* [9] used a free decay pendulum technique to measure the hydrodynamic coefficients directly.

This technique differs from the free-motion techniques mentioned above since the vehicle is mounted in a rig that constrains the vehicle motion. The method is based on the classical decay test with damped pendulum motion. A UUV is mounted to the pendulum rig and allowed to swing freely. It's planar and angular motion is captured on video which was then processed to determine Cartesian position coordinates, as well angular rate. This allowed for analytical computation of hydrodynamic parameters using the pendulum dynamics, and not the free-motion vehicle dynamics, limiting applicability to the problem at hand. Bahrke [19] used analytical as well as on-line statistical estimation

to estimate the speed, steering, and diving parameters of the equations of motion for the NPS AUV II vehicle. A Kalman filter based parameter estimator is used to estimate parameters.

Parametric models are informed by the physical characteristics of the system derived from first principles (e.g., the generic hydrodynamic model developed by Fossen [1]). The generic hydrodynamic model informs the basis functions required to model the system response, and the parameters define the relative weighting of these basis functions. Alternatively, generic sets of basis functions can be used to model the system dynamics. One such example is neural network representations, where, for example, radial basis functions are used. System identification techniques can similarly be used to learn the relative weights of these basis functions. These techniques are referred to as non-parametric approaches, a misnomer since the model is in fact parameterized, but these parameters do not have physical meaning. Non-parametric model representations for dynamical systems, such as neural networks, have been investigated (e.g., Scarselli and Tsoi [10]). According to Scarselli, a neural network with sufficient number of nodes is capable of learning and representing any dynamic system. These non-parametric models are less desirable since they lack physical meaning, but accurately capture complex input-output relations and thus find application is simulator development. A neural network approach has been applied to a UUV by Juan *et al.* [14]. The motion model of a Beaver underwater vehicle was developed using a wavelet neural network while the thruster model was developed using an improved radial basis function neural network.

C. SCOPE OF THIS WORK

This work develops a non-linear model, based on the rigid body dynamics and hydrodynamic force and moment analysis presented by Fossen, for a hovering-class UUV and learns the model parameters using various system identification techniques. This specialized model is first implemented in simulation using known coefficients (obtained from Chen *et al.* [5]), which is used to verify the system identification and online parameter estimation techniques (e.g., error convergence and parameter estimate

convergence to the true values, persistence of excitation, etc.). As a part of this analysis, the sequence of excitations is identified to learn the relevant model parameters. Next, these system identification techniques are applied to the physical system, the SeaBotix vLBV300, to obtain the parameters for the simplified hydrodynamic model. Position and orientation measurements, obtained from an instrumented dive tank environment and external motion capture system, are used to learn the dynamic model of the UUV in real-time. The predicted system response from the model is compared to the true response of the platform to validate the model. Similarly, a probabilistic estimation technique is evaluated for the system at hand. Finally, a non-parametric model is developed to evaluate applicability and limitations of this approach. The thesis is structured as follows:

Chapter II presents a brief history of the platform of interest, the SeaBotix vLBV300. An overview of its current and potential applications is presented as are appropriate vehicle specifications provided by the manufacturer.

Chapter III presents the development and specialization of the hydrodynamic (parametric) model for a hovering class UUV. The appropriate assumptions about the UUV and its assumed operating environment are presented as is the proposed reference frames. These assumptions allow for the simplification of the hydrodynamic model. Next, the appropriate and complete kinematic and dynamic equations of motion are developed and presented. Finally, a propulsion model is developed for the individual thrusters as well as the vectored thruster system and presented.

Chapter IV discusses the various system identification techniques investigated. Recursive methods such as prediction error based techniques using gradient estimators or least-squared estimators are investigated as are a statistical approach based on Bayesian Inferencing, and a non-parametric approach based on neural networks.

Chapter V presents the results of a simulated parameter estimation experiment based on known parameters from previous research [5]. The framework for the parametric model derived in Chapter III and the system identification techniques derived in Chapter IV are validated and results presented. Also presented are the parameter

estimation results and a complete hydrodynamic model for the vLBV300 vehicle developed from real world implementation of the techniques discussed in Chapters III and IV. This chapter also presents a verification of the results as well as sequential procedure for determining the dynamic coefficients of a UUV using on-line parameter estimation based on conclusions obtained from the experimental data

Chapter VI presents conclusions based on the results and limitations identified by this work. It also presents areas for further refinement or research and conclusions on this work.

THIS PAGE INTENTIONALLY LEFT BLANK

II. SEABOTIX VLBV300 REMOTELY OPERATED VEHICLE

A. OVERVIEW

The vLBV300, manufactured by SeaBotix, Inc. of San Diego, California, is a MiniROV/Hovering class remotely operated vehicle. It is operated through a tether and can be controlled with a joystick. Data and power are transmitted through the tethered interface. For propulsion and maneuvering the vLBV300 uses six brushless DC thrusters, four of which are vectored in the horizontal plane (surge, sway, yaw), the angles of which can be manually adjusted, while the remaining two are fixed in the vertical plane (heave, roll). As a result, the vehicle allows control in 5 degrees of freedom (DOF) when individual thrusters are commanded, or 4 DOF when controlled via the joystick. The vehicle can be equipped with a variety of sensors including a controllable HD camera, rear and side cameras, sonars, as well as a grabber arm for in-water intervention. A typical configuration is shown in Figure 1.

Figure 1. SeaBotix vLBV300 miniROV platform (From [15]) has been modified to allow tethered, autonomous operations.

Based on its specifications, SeaBotix recommends the vLBV300 for a variety of applications including work on offshore oil or gas platforms, coastal and inshore surveys, maritime security of ports, harbors and vessels, or long line penetration or pipe

inspections. The vehicle is operated in tele-operated mode for these applications: a human pilot controls the vehicle via the joystick in the surge, sway, heave, and yaw directions.

In the current research, a different application is investigated: *autonomous* close-proximity operations in the presence of static and dynamic obstacles (i.e., human divers). The vLBV300 is particularly well suited to proximal operations for the following reasons:

- The vehicle has 5 degrees of freedom, including surge, sway, heave, yaw, and roll. Controllability in the sway direction is of particular importance for proximal operations to ensure diver safety.
- The vehicle is light-weight (i.e., easy to deploy and recover), neutrally buoyant (to ensure safe operations among divers), and powerful enough to maneuver among divers and carry considerable payloads.
- The vehicle has an open-frame architecture, allowing various payloads to be integrated with relative ease.
- A computer control interface has been developed in addition to the joystick interface that allows autonomous operation of the vehicle. A high-level control interface allows computer control via the joystick commands, while a low-level control interface allows individual thrusters to be commanded.

B. VEHICLE SPECIFICATIONS

Specifications for the vehicle are provided by the manufacturer. Table 1 lists general specification of the vLBV300 and Table 2 lists tether specifications. The specific vehicle setup for the scope of this work is discussed in the next section.

Depth Rating	304.8 m
Length	0.625 m
Width	0.39 m
Height	0.39 m
Diagonal	0.55 m
Weight in air	18.09 kg

Table 1. General characteristics (From [15])

Diameter	0.889 cm
Length	250 m (nominal)
Working load	978.6 N
Breaking strength	6.86 kN
Buoyancy	neutral

Table 2. Tether characteristics (From [15])

C. CAVR VEHICLE CONFIGURATION

The vLBV300 is a miniROV remotely operated vehicle which can be commanded via a high- or low-level computer interface, resulting in a tethered, hovering-class autonomous underwater system (THAUS). As configured for this research, the vehicle weighs 20.9 kg in air; is 0.625 m long, 0.39 m wide, and 0.39 m tall; and has a BlueView sonar attached in addition to the standard payload (tilt-controlled forward-looking camera and LED arrays). This payload will be expanded with an inertial navigation system (INS), Doppler velocity log (DVL) and GPS unit in the near future.

The aft pair of horizontal thrusters is vectored to an angle of 45 degrees from centerline while the forward pair of horizontal thrusters is vectored to 35 degrees from centerline. The vertical thrusters are angled at 18 degrees (see Figure 2).

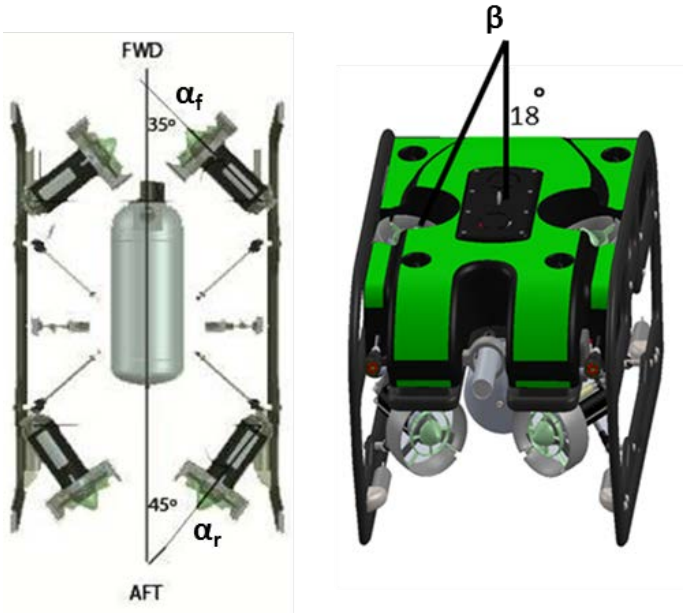

Figure 2. Horizontal and vertical thruster configuration of the vLBV300

A right handed, body-fixed frame is used with *x* pointing forward, *y* pointing right, and *z* pointing down (see Figure (3)).

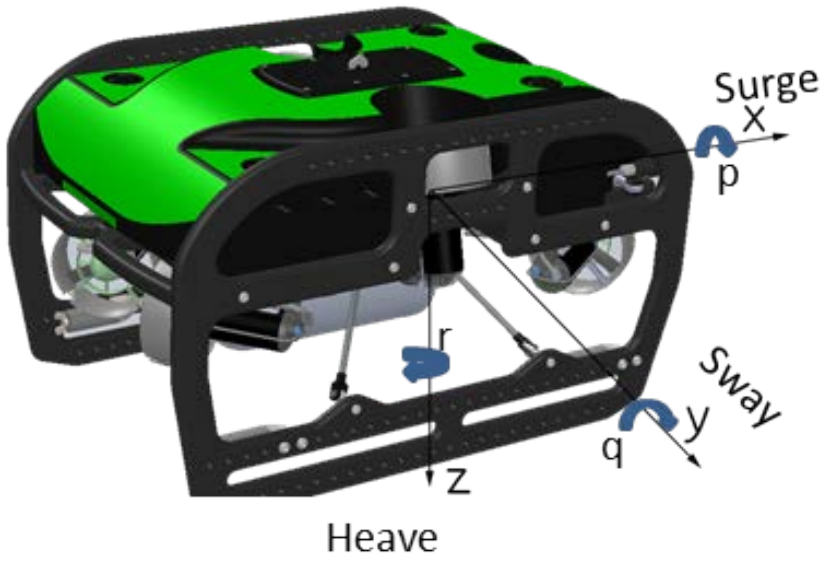

Figure 3. Body reference frame of the vLBV300

D. PROPULSION MODEL

As discussed in Section B, the SeaBotix ROV can be controlled either via a high-level, joystick interface that allows for four DOF control or via low-level pulse width modulation (PWM) thruster commands directly to each thruster via a computer program which provides for five DOF control. In order to ensure maximum control of the vehicle, for this research only the low-level commands are used.

The low-level PWM commands range from -102 to 102 and is related to the thruster RPM. For the aft thruster-pair, the positive direction is defined as producing clockwise rotation of the propeller, which results in motion in the positive (ahead) surge direction. For the forward thrusters, a positive command results in counter-clockwise rotation and therefore also thrust in the positive surge direction. A positive sway command results in the leading edge propellers (right hand set for motion to the right) spinning in the positive direction while the trailing set spins in the negative direction. A positive command to the vertical thrusters results in clockwise rotation and motion in the

positive heave (downward) direction. Using these definitions and the thruster geometry, the vehicle's propulsion forces are defined below. For ease of notation the thrust generated by the horizontal, forward, starboard propeller is referred to here as *FS*. The horizontal, forward, port propeller thrust is referred to as *FP*. The horizontal, aft, starboard propeller thrust is referred to as *AS* and the horizontal, aft, port propeller thrust is referred to as *AP*. The thrust generated by the vertical starboard and vertical port propellers are referred to as *VS* and *VP*, respectively. The vectoring angle for the forward pair of thrusters is called α_f, the vectoring angle for the aft pair of thrusters is called α_r, and the vectoring angle for the vertical pair of thrusters is called β (see Figure 2). X_{prop} is the total force generated by all thrusters in the surge direction, Y_{prop} is the total force generated by all thrusters in the sway direction, and Z_{prop} is the total force generated by all thrusters in the heave direction. K_{prop} is the total moment generated around the body *x*-axis, M_{prop} is the total moment generated around the body *y*-axis, and N_{prop} is the total moment generated around the body *z*-axis.

$$X_{prop} = \left(AP * \mathrm{c}\,\alpha_r - FP * \mathrm{c}\,\alpha_f\right) + \left(AS * \mathrm{c}\,\alpha_r - FS * \mathrm{c}\,\alpha_f\right) \tag{4}$$

$$Y_{prop} = \left(FS * \mathrm{s}\,\alpha_f + AS * \mathrm{s}\,\alpha_r\right) - \left(FP * \mathrm{s}\,\alpha_f + AP * \mathrm{s}\,\alpha_r\right) - (VL - VR) * \mathrm{s}\,\beta \tag{5}$$

$$Z_{prop} = (VL + VR) * \mathrm{c}\,\beta \tag{6}$$

$$K_{prop} = (VL + VR) * \mathrm{s}\,\beta * LV \tag{7}$$

$$M_{prop} = 0 \tag{8}$$

$$
\begin{aligned}
N_{prop} = & \left(AP * \mathrm{c}\,\alpha_r - FP * \mathrm{c}\,\alpha_f \right) * LH_1 + \\
& \left(AS * \mathrm{c}\,\alpha_r - FS * \mathrm{c}\,\alpha_f \right) * LH_1 + \\
& \left(AP * \mathrm{s}\,\alpha_r - FP * \mathrm{s}\,\alpha_f \right) * LH_2 + \\
& \left(AS * \mathrm{s}\,\alpha_r - FS * \mathrm{s}\,\alpha_f \right) * LH_2 +
\end{aligned}
\tag{9}
$$

Note: For ease of notation s● = sin(●), c●=cos(●), and t●=tan(●). *LV*, LH_1, and LH_2 are moment arms. *LV* is defined as the distance between the centroid of the two vertical propellers while LH_1 is the distance between the centroid of the propellers of the aft or forward pair, and LH_2 is defined as the distance between the centroid of the propellers of the right or left side pair. M_{prop} is zero in all cases since pitch cannot be controlled on any vehicle considered in this research.

The relationship between low-level PWM commands and a measured force for individual thrusters is determined through in-tank testing using the aft thruster pair only and a Futek USB210 strain gage. The results for both the positive and negative direction are virtually identical for commands from 10 to 60, as would be expected from the performance of a fixed pitch propeller. Due to a limitation with the experimental setup, surface effects affected results above a PWM command of 60 (i.e., the 60-102 range) as can be seen in Figure 4. The vehicle was secured to an anchor point outside the tank via a line (with the strain gage integrated into this line) in order to keep the strain gage dry. Because the line rested on the lip of the tank which is necessarily several inches higher than the water line, the vehicle assumed a negative pitch (downward) angle during every test. This resulted in the aft pair of thrusters being up to several inches closer to the surface at higher PWM commands than the forward thrusters. This difference in depth, and therefore water pressure, was significant enough above the PWM command of 60 for the aft pair of thrusters to experience significant cavitation, reducing their effectiveness. The aft pair of thrusters was used for the positive direction testing and the forward pair of thrusters was used for the negative direction testing.

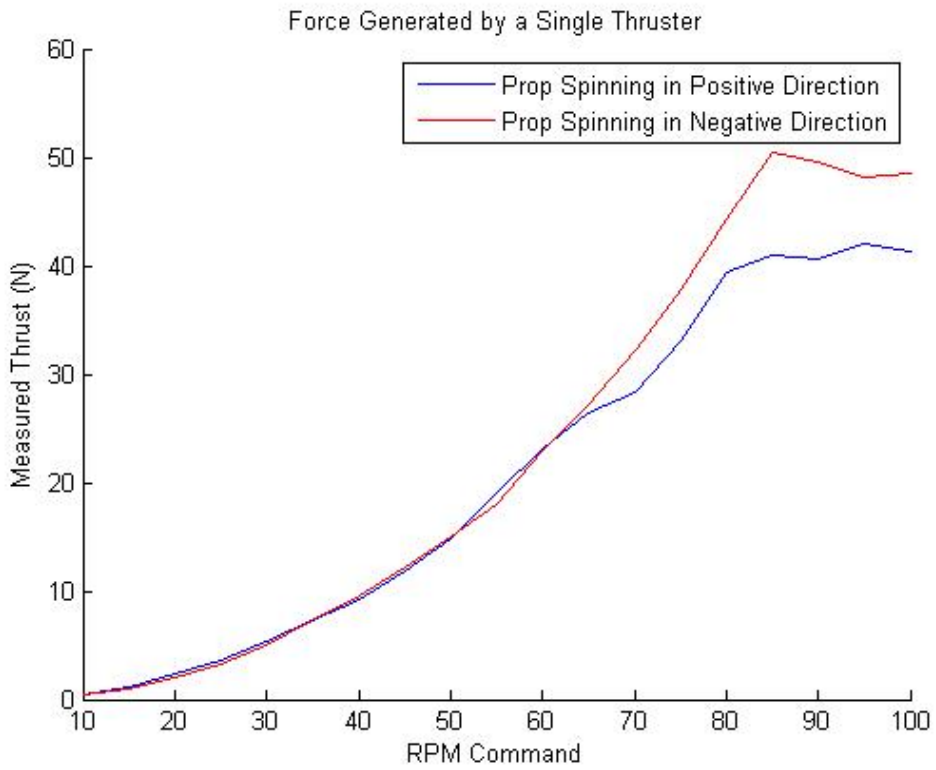

Figure 4. Thruster testing showing surface effects

Because of this, a regression was only fit to data for PWM commands between 10 and 60. Since the planned vehicle operations will be at low thrust (and speed), this thruster model is sufficient. The truncated, experimentally obtained data was then used to fit a second-order polynomial to relate PWM values to generated thrust.

$$Thrust = 0.006736PWM^2 - 0.03366PWM + 0.0684 \tag{10}$$

The results of the regression are compared with the experimental data in Figure 5.

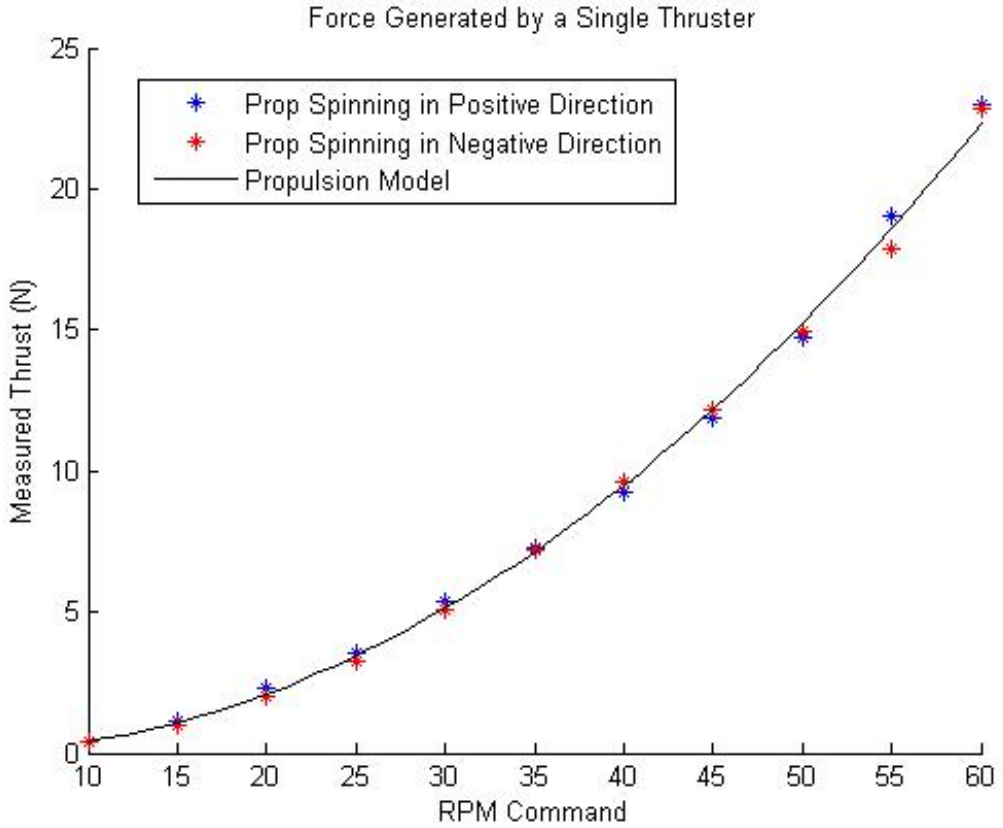

Figure 5. Bounded propulsion model for a single thruster on the vLBV300 compared with experimental data

III. GENERIC MOTION MODEL

A. FULL EQUATIONS OF MOTION

The non-linear dynamic equations of motion for a six degree of freedom (DOF) submersed body are presented by Fossen [1] as

$$M\dot{\upsilon} + C(\upsilon)\upsilon + D(\upsilon)\upsilon + g(\eta) = \tau \tag{11}$$

The vector υ contains the body velocities and angular rates,

$$\upsilon = [u, v, w, p, q, r]^T \tag{12}$$

where u is velocity in the surge direction, v is velocity in the sway direction, w is velocity in the heave direction, p is the angular rate about the x-axis of the body frame, q is the angular rate about the y-axis of the body frame, and r is the angular rate about the z-axis of the body. All velocities in this work are presented in units of meters per second and all angular rates in radians per second.

The vehicle's pose (position and orientation) is described by the vector η:

$$\eta = [x, y, z, \phi, \theta, \psi]^T \tag{13}$$

where x, y, and z are the vehicle's position with respect to an inertial reference frame, and ϕ, θ, ψ are Euler angles determined using the *zyx*-convention. All positions in this work are presented in units of meters and all Euler angles in units of radians.

Vehicle pose dynamics are related to body velocities, angular rates, and the pose itself, by

$$\dot{\eta} = J(\eta)\upsilon \tag{14}$$

$$J = \begin{bmatrix} J_1 & 0_{3\times3} \\ 0_{3\times3} & J_2 \end{bmatrix} \tag{15}$$

where

$$J_1 = \begin{bmatrix} \mathrm{c}\psi\,\mathrm{c}\theta & -\mathrm{s}\psi\,\mathrm{c}\phi + \mathrm{c}\psi\,\mathrm{s}\theta\,\mathrm{s}\phi & \mathrm{s}\psi\,\mathrm{s}\phi + \mathrm{c}\psi\,\mathrm{c}\phi\,\mathrm{s}\theta \\ \mathrm{s}\psi\,\mathrm{c}\theta & \mathrm{c}\psi\,\mathrm{c}\phi + \mathrm{s}\phi\,\mathrm{s}\theta\,\mathrm{s}\psi & -\mathrm{c}\psi\,\mathrm{s}\phi + \mathrm{s}\theta\,\mathrm{s}\psi\,\mathrm{c}\phi \\ -\mathrm{s}\theta & \mathrm{c}\theta\,\mathrm{s}\phi & \mathrm{c}\theta\,\mathrm{c}\phi \end{bmatrix} \tag{16}$$

$$J_2 = \begin{bmatrix} 1 & s\theta t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi / c\theta & c\phi / c\theta \end{bmatrix} \tag{17}$$

Matrices M and $C(\upsilon)$ are each composed of two parts: a contribution from the rigid-body dynamics and a contribution from the added, or apparent, mass and inertia. The rigid body dynamics are derived by Fossen [1] based on Newton's second law and Euler's first and second axioms to that law while the added mass and inertia terms are derived from an energy based approach resulting from Kirchoff's equations as discussed in [1].

The inertia matrix, M, is the sum of the rigid body mass, M_{RB} , and the added inertia matrix, M_A,

$$M \triangleq M_{RB} + M_A \tag{18}$$

$$M_{RB} = \begin{bmatrix} m & 0 & 0 & 0 & mz_g & -my_g \\ 0 & m & 0 & -mz_g & 0 & mx_g \\ 0 & 0 & m & my_g & -mx_g & 0 \\ 0 & -mz_g & my_g & I_X & -I_{XY} & -I_{XZ} \\ mz_g & 0 & -mx_g & -I_{YX} & I_Y & -I_{YZ} \\ -my_g & mx_g & 0 & -I_{ZX} & -I_{ZY} & I_Z \end{bmatrix} \tag{19}$$

$$M_A = \begin{bmatrix} X_{\dot{u}} & X_{\dot{v}} & X_{\dot{w}} & X_{\dot{p}} & X_{\dot{q}} & X_{\dot{r}} \\ Y_{\dot{u}} & Y_{\dot{v}} & Y_{\dot{w}} & Y_{\dot{p}} & Y_{\dot{q}} & Y_{\dot{r}} \\ Z_{\dot{u}} & Z_{\dot{v}} & Z_{\dot{w}} & Z_{\dot{p}} & Z_{\dot{q}} & Z_{\dot{r}} \\ K_{\dot{u}} & K_{\dot{v}} & K_{\dot{w}} & K_{\dot{p}} & K_{\dot{q}} & K_{\dot{r}} \\ M_{\dot{u}} & M_{\dot{v}} & M_{\dot{w}} & M_{\dot{p}} & M_{\dot{q}} & M_{\dot{r}} \\ N_{\dot{u}} & N_{\dot{v}} & N_{\dot{w}} & N_{\dot{p}} & N_{\dot{q}} & N_{\dot{r}} \end{bmatrix} \tag{20}$$

where m is the vehicle mass in air and x_g, y_g, and z_g are the Cartesian coordinates in the body frame of the vehicle's center of gravity. I_X is the vehicle's moment of inertia around the x-axis of the body, I_Y is the vehicle's moment of inertia around the y-axis of the body, and I_Z is the vehicle's moment of inertia around the z-axis of the body. These terms represent the moment of inertia due to un-coupled, single DOF rotation about the axis of the DOF movement. The inertia terms with more than one letter in the subscript

denote the vehicle's moment of inertia due to coupled motion. For example I_{XY} denotes the vehicle's moment of inertia around the vehicle's *x*-axis due to rotation around the *y*-axis.

Fossen defines the added mass coefficients shown in Equation (20) as describing the pressure-induced forces and moments due to a forced harmonic motion of the body which are proportional to the acceleration of the body [1]. For example $X_{\dot{v}}$ is the force along the x-axis due to an acceleration, $\dot{v}$, in the y-direction. Mathematically, this is expressed as

$$X_{\dot{v}} \triangleq \frac{\partial X}{\partial \dot{v}} \tag{21}$$

C(υ) is the matrix of Coriolis and centripetal terms and is also the sum of a rigid body, C_{RB}, and added mass, C_A, components:

$$C(\upsilon) \triangleq C_{RB}(\upsilon) + C_A(\upsilon) \tag{22}$$

where

$$C_{RB}(\upsilon) = \begin{bmatrix} 0_{3\times 3} & C_1 \\ C_2 & C_3 \end{bmatrix} \tag{23}$$

and

$$C_1 = \begin{bmatrix} m(y_g q + z_G r) & -m(x_g q - w) & -m(x_g r + v) \\ -m(y_g p + w) & m(z_G r + x_g p) & -m(y_g r - u) \\ -m(z_G p - v) & -m(z_G q + u) & m(x_g p + y_g q) \end{bmatrix} \tag{24}$$

$$C_2 = \begin{bmatrix} -m(y_g q + z_G r) & m(y_g p + w) & m(z_G p - v) \\ m(x_g q - w) & -m(z_G r + x_g p) & m(z_G q + u) \\ m(x_g r + v) & m(y_g r - u) & -m(x_g p + y_g q) \end{bmatrix} \tag{25}$$

$$C_3 = \begin{bmatrix} 0 & -I_{YZ} q - I_{XZ} p + I_Z r & I_{YZ} r + I_{XY} p - I_Y q \\ I_{YZ} q + I_{XZ} p - I_Z r & 0 & -I_{XZ} r - I_{XY} q + I_X p \\ -I_{YZ} r - I_{XY} p + I_Y q & I_{XZ} r + I_{XY} q - I_X p & 0 \end{bmatrix} \tag{26}$$

$$C_A(\upsilon) = \begin{bmatrix} 0 & 0 & 0 & 0 & -a_3 & a_2 \\ 0 & 0 & 0 & a_3 & 0 & -a_1 \\ 0 & 0 & 0 & -a_2 & a_1 & 0 \\ 0 & -a_3 & a_2 & 0 & -b_3 & b_2 \\ a_3 & 0 & -a_1 & b_3 & 0 & -b_1 \\ -a_2 & a_1 & 0 & -b_2 & b_1 & 0 \end{bmatrix} \tag{27}$$

where

$$\begin{aligned} a_1 &= X_{\dot{u}}u + X_{\dot{v}}v + X_{\dot{w}}w + X_{\dot{p}}p + X_{\dot{q}}q + X_{\dot{r}}r \\ a_2 &= X_{\dot{v}}u + Y_{\dot{v}}v + Y_{\dot{w}}w + Y_{\dot{p}}p + Y_{\dot{q}}q + Y_{\dot{r}}r \\ a_3 &= X_{\dot{w}}u + Y_{\dot{w}}v + Z_{\dot{w}}w + Z_{\dot{p}}p + Z_{\dot{q}}q + Z_{\dot{r}}r \\ b_1 &= X_{\dot{p}}u + Y_{\dot{p}}v + Z_{\dot{p}}w + K_{\dot{p}}p + K_{\dot{q}}q + K_{\dot{r}}r \\ b_2 &= X_{\dot{q}}u + Y_{\dot{q}}v + Z_{\dot{q}}w + K_{\dot{q}}p + M_{\dot{q}}q + M_{\dot{r}}r \\ b_3 &= X_{\dot{r}}u + Y_{\dot{r}}v + Z_{\dot{r}}w + K_{\dot{r}}p + M_{\dot{r}}q + N_{\dot{r}}r \end{aligned} \tag{28}$$

The damping matrix, *D(υ)*, is the sum of radiation-induced potential damping due to forced body oscillation, D_P, linear skin friction due to laminar boundary flow and quadratic skin friction due to turbulent flow, D_S, wave drift damping, D_W, and damping due to vortex shedding, D_M,

$$D(\upsilon) \triangleq D_P(\upsilon) + D_S(\upsilon) + D_W(\upsilon) + D_M(\upsilon) \tag{29}$$

Radiation induced damping, $D_{P,}$ is often referred to as potential damping and is a result of a body being forced to oscillate with the excitation frequency of waves encountered which results in added mass, damping, and restoring forces. While generally negligible compared to other forces at great depth for underwater vehicles, it is of more concern for surface vehicles.

Damping due to skin friction is a function of the vehicle's exterior make up as well as speed. Low-speed, laminar flow results in a low frequency contribution while turbulent flow results in a high frequency contribution.

Wave drift damping is only of significance for surface vessels advancing into waves. Damping due to vortex shedding occurs in a non-viscous fluid and is a function of the speed at which the vehicle moves, the density of the water in which it is operating, the

projected cross sectional area and the vehicle's Reynolds number. This is the most significant of the damping contributors for an underwater vehicle. The general equation for this damping is further explored by Equation (45). g(η) is the vector of restorative forces and moments,

$$g(\eta) = \begin{bmatrix} (W-B)\mathrm{s}\theta \\ -(W-B)\mathrm{c}\theta\mathrm{s}\phi \\ -(W-B)\mathrm{c}\theta\mathrm{c}\phi \\ -(y_G W - y_B B)\mathrm{c}\theta\mathrm{c}\phi + (z_G W - z_B B)\mathrm{c}\theta\mathrm{s}\phi \\ (z_G W - z_B B)\mathrm{s}\theta + (x_G W - x_B B)\mathrm{c}\theta\mathrm{c}\phi \\ -(x_G W - x_B B)\mathrm{c}\theta\mathrm{s}\phi - (y_G W - y_B B)\mathrm{s}\theta \end{bmatrix} \tag{30}$$

where W is the weight (in air) of the vehicle, $W = mg$, B is buoyant force produced by the fully submerged vehicle, and x_g, y_g, and z_g are the Cartesian coordinates in the body frame of the vehicle's center of gravity while x_b, y_b, and z_b are the Cartesian coordinates in the body frame of the vehicle's center of buoyancy. All gravitational forces and moments are assumed to act through and around the vehicle's center of gravity while all buoyancy forces and moments are assumed to act through and around the vehicle's center of buoyancy.

The vector of the forces and moments exerted on the vehicle by the thrusters is τ. Recall from Equations (4) - (9) that X_{prop}, Y_{prop}, and Z_{prop} are forces in the translational DOF while K_{prop}, M_{prop}, and N_{prop} are moments in the rotational DOF.

$$\tau = \begin{bmatrix} X_{prop} & Y_{prop} & Z_{prop} & K_{prop} & M_{prop} & N_{prop} \end{bmatrix} \tag{31}$$

The sign convention for the forces that make up τ follow the right handed reference frame sign convention established for the vehicle in Chapter II. That is, X_{prop} is positive when it generates a force in the positive surge direction (forward). The force Y_{prop} is positive when it generates a force in the positive sway direction (right). The force Z_{prop} is positive when it generates a force in the positive heave direction (downward). The moment K_{prop} is positive when it generates a right-handed moment around the x-axis. The moment M_{prop} is positive when it generates a right-handed moment around the y-axis. The moment N_{prop} is positive when it generates a right handed moment around the z-axis.

B. ASSUMPTIONS AND SIMPLIFIED EQUATIONS

In order to model the non-linear dynamics of the UUV, Equations (11) and (14) must be solved for υ and η. Several assumptions are made about the characteristics of the UUV to simplify the components of Equation(11). The vehicle is assumed to be neutrally buoyant ($B=W$) and the tether is assumed to have no effect on vehicle motion or trim. The x and y coordinates of the center of gravity and center of buoyancy are assumed to coincide. Also, since the specific mass distribution of the vehicle is not known, a uniform distribution of mass is assumed [5, 6] and the components of the inertia tensor that are off the main diagonal are neglected. The moments of inertia are estimated using the equation for moment of inertia of a solid, rectangular cuboid

$$I_X = \frac{1}{12} m(height^2 + width^2) \tag{32}$$

$$I_Y = \frac{1}{12} m(height^2 + length^2) \tag{33}$$

$$I_Z = \frac{1}{12} m(length^2 + width^2) \tag{34}$$

and result in I_X = 0.559 kgm^2, I_Y = 0.953 kgm^2, and I_Z = 0.90 kgm^2 for the vLBV300. These assumptions further simplify the equations of motion to yield:

$$g(\eta) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -B(z_G - z_B)\mathrm{c}\theta\,\mathrm{s}\phi \\ B(z_G - z_B)\mathrm{s}\theta \\ 0 \end{bmatrix} \tag{35}$$

$$F_{BK} = B(z_G - z_B)\mathrm{c}\theta\,\mathrm{s}\phi \tag{36}$$

$$F_{BM} = B(z_G - z_B)\mathrm{s}\theta \tag{37}$$

$$M_{RB} = \begin{bmatrix} m & 0 & 0 & 0 & mz_G & 0 \\ 0 & m & 0 & -mz_G & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & -mz_G & 0 & I_X & 0 & 0 \\ mz_G & 0 & 0 & 0 & I_Y & 0 \\ 0 & 0 & 0 & 0 & 0 & I_Z \end{bmatrix} \tag{38}$$

$$C_{RB}(\upsilon) = \begin{bmatrix} 0_{3\times 3} & C_1 \\ C_2 & C_3 \end{bmatrix} \tag{39}$$

where

$$C_1 = \begin{bmatrix} mz_G r & mw & -mv \\ -mw & mz_G r & mu \\ -m(z_G p - v) & -m(z_G q + u) & 0 \end{bmatrix} \tag{40}$$

$$C_2 = \begin{bmatrix} -mz_G r & mw & m(z_G p - v) \\ -mw & -mz_G r & m(z_G q + u) \\ mv & -mu & 0 \end{bmatrix} \tag{41}$$

$$C_3 = \begin{bmatrix} 0 & I_Z r & -I_Y q \\ -I_Z r & 0 & I_X p \\ I_Y q & -I_X p & 0 \end{bmatrix} \tag{42}$$

Furthermore, the vehicle is assumed to have three planes of symmetry, will be fully submerged for all motion, and will only move at low speeds. Therefore, (see [1] Equations (2.17) and (2.23)) the following simplified terms are obtained:

$$M_A = -\text{diag}\left(\left[X_{\dot{u}}, Y_{\dot{v}}, Z_{\dot{w}}, K_{\dot{p}} M_{\dot{q}}, N_{\dot{r}} \right]\right) \tag{43}$$

$$C_A(\upsilon) = -\begin{bmatrix} 0 & 0 & 0 & 0 & -Z_{\dot{w}} w & Y_{\dot{v}} v \\ 0 & 0 & 0 & Z_{\dot{w}} w & 0 & -X_{\dot{u}} u \\ 0 & 0 & 0 & -Y_{\dot{v}} v & X_{\dot{u}} u & 0 \\ 0 & -Z_{\dot{w}} w & Y_{\dot{v}} v & 0 & -N_{\dot{r}} r & M_{\dot{q}} q \\ Z_{\dot{w}} w & 0 & -X_{\dot{u}} u & N_{\dot{r}} r & 0 & -K_{\dot{p}} p \\ -Y_{\dot{v}} v & X_{\dot{u}} u & 0 & -M_{\dot{q}} q & K_{\dot{p}} p & 0 \end{bmatrix} \tag{44}$$

Assuming in addition that the vehicle is performing only uncoupled motion with complete turbulent flow, the damping term is in the form:

$$D(\upsilon) = -\text{diag}\left(\left[X_{u|u|}|u|, Y_{v|v|}|v|, Z_{w|w|}|w|, K_{p|p|}|p|, M_{q|q|}|q|, N_{r|r|}|r| \right]\right) \tag{45}$$

Combining all assumptions yields the following simplified dynamic equations for non-coupled motion for the THAUS UUV:

$$\begin{aligned}
X_{prop} &= (m - X_{\dot{u}})\dot{u} - (m - Y_{\dot{v}})vr + (m - Z_{\dot{w}})wq - X_{u|u|}u|u| \\
Y_{prop} &= (m - Y_{\dot{v}})\dot{v} - (m - Z\dot{w})wp + (m - X_{\dot{u}})ur - Y_{v|v|}v|v| \\
Z_{prop} &= (m - Z_{\dot{w}})\dot{w} - (m - X_{\dot{u}})uq + (m - Y_{\dot{v}})vp - Z_{w|w|}w|w| \\
K_{prop} &= (I_X - K_{\dot{p}})\dot{p} + (Y_{\dot{v}} - Z_{\dot{w}})wv + (I_Z - I_Y + N_{\dot{r}} + M_{\dot{q}})rq \\
&\quad -K_{p|p|}p|p| - F_{BK} \\
M_{prop} &= (I_Y - M_{\dot{q}})\dot{q} + (Z_{\dot{w}} - X_{\dot{u}})uw + (I_X - I_Z + N_{\dot{r}} - K_{\dot{p}})rp \\
&\quad -M_{q|q|}q|q| + F_{BM} \\
N_{prop} &= (I_Z - N_{\dot{r}})\dot{r} + (X_{\dot{u}} - Y_{\dot{v}})uv + (I_Y - I_X + K_{\dot{p}} - M_{\dot{q}})rp \\
&\quad -N_{r|r|}r|r|
\end{aligned} \tag{46}$$

The validity of these assumptions, based on the accuracy of the learned model compared to the measured response, will be discussed in Chapter VI.

IV. SYSTEM IDENTIFICATION

A. PARAMETER ESTIMATION FOR A STATIC SYSTEM

Consider a regression model of the form [8]

$$y_i = \varphi_i^1 \theta^1 + \varphi_i^2 \theta^2 + ... + \varphi_i^m \theta^m = \varphi_i^T \theta \tag{47}$$

where y_i, $i=1,2,\ldots,n$, is a series of observations, $\varphi_i = \begin{bmatrix} \varphi_i^1 & \varphi_i^2 & ... & \varphi_i^m \end{bmatrix}^T$ are the known regressor functions (which are functions of the states and controls), and $\Theta = \begin{bmatrix} \theta^1 & \theta^2 & ... & \theta^m \end{bmatrix}^T$ are the unknown parameters (to be estimated). The goal of online system identification is to recursively estimate the unknown parameters Θ according to some adaptation law as new information becomes available.

Several methods of parameter estimation are available to perform the necessary recursive estimation, including the gradient estimator, the recursive linear least squares (RLLS) estimator, a statistical, Bayesian approach, and a neural network based approach.

1. Recursive Linear Least Squares

The goal of the linear least squares estimator is to minimize the square of the error between the complete history of the measured and modeled response of the system with respect to Θ, as in [2] and [8].

$$\min_{\Theta} V(\Theta, n) = \min_{\Theta} \frac{1}{2} \sum_{i=1}^{n} (y_i - \varphi_i^T \Theta)^2 \tag{48}$$

To accomplish this, let $Y = \begin{bmatrix} y_1 & y_2 & ... & y_n \end{bmatrix}^T$ be the set of measurements and define the error terms $\varepsilon_i = y_i - \varphi_i^T \hat{\Theta}$, where $\hat{\Theta}$ is an estimate of the unknown parameter vector. This can be written in vector notation: $E = \begin{bmatrix} \varepsilon_1 & \varepsilon_2 & ... & \varepsilon_n \end{bmatrix}^T$. Finally, let

$$\Phi = \begin{bmatrix} \varphi_1^T \\ \vdots \\ \varphi_n^T \end{bmatrix} \tag{49}$$

Then the loss function in Equation (48) can be written as:

$$V(\theta, n) = \frac{1}{2}\sum_{i=1}^{n} \varepsilon_i^2 = \frac{1}{2} E^T E = \frac{1}{2}\left\|E\right\|^2 \tag{50}$$

where $E = Y - \Phi\Theta$. This loss function is minimized by:

$$\hat{\Theta}^* = \left(\Phi^T \Phi\right)^{-1} \Phi^T Y \tag{51}$$

which requires that $\Phi^T\Phi$ is non-singular to yield a unique solution. In a similar manner, the recursive linear least square (RLLS) estimator is defined as

$$\hat{\Theta}_i = \hat{\Theta}_{i-1} + K_i(y_i - \varphi_i^T \hat{\Theta}_{i-1}) \tag{52}$$

where the filter gain is given by:

$$K_i = P_{i-1}\varphi_i \left(I + \varphi_i^T P_{i-1}\varphi_i\right)^{-1} \tag{53}$$

and $P_i = (I - K_i\varphi_i^T)P_{i-1}$. To ensure a unique solution to RLLS estimation $\Phi^T\Phi = \sum_{i=1}^{k} \varphi_i\varphi_i^T$ must be of full rank, or non-singular.

2. Neural Network

Computational neural networks (NN) are modeled after the understanding of biological nervous systems and modeled as a dense web of elements or nodes with weighted connections that may be adapted, or trained, during use of the network to improve performance [16]. The strength of the NN lies in its ability to process many computational tasks in parallel instead of in a serial manner as done by most computers, in its inherent robustness, and in its ability for adaptation or learning. The large number of interconnected nodes allows for several competing hypotheses to be investigated and pursued simultaneously by the network. Because a NN is composed of a large number of interconnected nodes, damage to one, or even several, may not completely destroy the system as would happen in a serial computer. Neural networks using variable weights between nodes and a learning strategy also have the ability to learn and improve performance during operation or after damage. This ability also provides for a degree of robustness in the presence of variability of the processing element.

Neural networks are of interest to this work because of this ability to learn and adapt online as well as their ability to accurately model complicated non-linear systems. While the end results is a non-parametric "black box" input-output model that does not afford the same physical understanding of a system as the hydrodynamic model by Fossen does, the NN model can provide a highly accurate simulation that is able to account for the nonlinearities inherent to modeling an underwater vehicle.

The simplest type of neural network sums N inputs of x_i which commonly represents a vehicle's state or control input such as thruster commands. x_i is multiplied by its respective weights, w_i, and the network passes the results through a non-linear function, f, with a delay described by θ as presented by Lippmann [16]. The non-linear function is called the "basis" function and helps define the architecture of the model in the same manner as the regressor does in a parametric representation. A visual representation is given in Figure 6.

$$y = f\left(\sum_{i=0}^{N-1} w_i x_i - \theta\right) \tag{54}$$

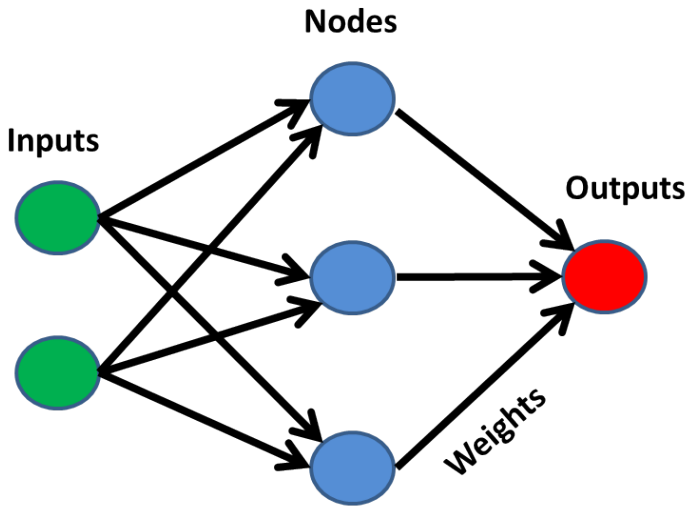

Figure 6. A basic neural network showing the network output expressed as a sum of weighted inputs.

The goal of a large group of nodes becomes to determine which of a large set of functions is best representative of an unknown input function or series of input functions,

as well as non-parametric uncertainties. Non-parametric uncertainties typically encompass measurement noise, equipment failure, environmental disturbances or other uncertainties [21].

In system identification terms, as presented in Section VI.A, this is expressed as:

$$y_i = \theta^T \varphi_i + \varepsilon_i \tag{55}$$

where ε_i represents the approximation error of the non-parametric uncertainties. In order to identify the parameters contained in θ^T in the face of these uncertainties, a set of basis functions, φ_i, which in the RLLS context were informed by the parametric model itself, must be chosen. These basis functions are chosen such that the error ε_i becomes small over the expected area of operation of the system being identified.

Physically, this means that a series of basis functions are tested, along with weights for each portion of the basis function until the basis function set and associated weights are found that most closely represents the uncertain input-output relation. Lippmann presents an adaptive classifier along these lines (see Figure 7) [16] that takes a series of inputs and computes a series of performance related scores that reflect the accuracy of each basis function examined. These scores are passed on to a second stage where only the best score is passed. This best score is then used to adapt the weights of the network.

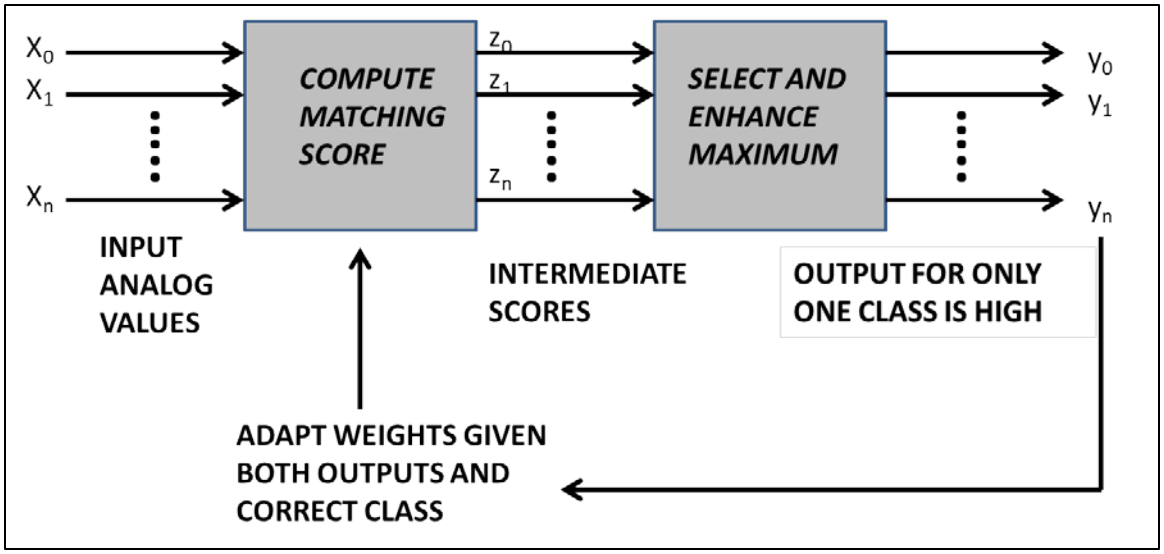

Figure 7. Diagram of an adaptive neural network (After [16])

"Training," or adaptation of the network, can take place online as in Figure 7, or prior to using the network. Input data is fed to the network in concert with known output data. The network then trains itself according to various methods and a set of coefficients are produced. Two of the most well-known adaptation schemes are the steepest decent algorithm, or error back propagation (EBP), and the Gauss-Newton algorithm. The EBP method is presented in [17] as

$$w_{i+1} = w_i + \alpha g_i \tag{56}$$

where w_i is the weight assigned to a particular node at step i, α is the step size, g_i is the first-order derivative of the total error function $E(x, w)$, and w_{i+1} is the updated node weighting. Further notation in [17] includes: p is the index of training patterns from 1 to P, m is the index of outputs from 1 to M, and N is the total number of weights.

The Gauss-Newton method uses the Jacobian matrix, J, to relate the gradient g to a vector of errors e.

$$g = Je \tag{57}$$

$$J = \begin{bmatrix} \frac{\partial e_{1,1}}{dw_1} & \frac{\partial e_{1,1}}{dw_2} & \cdots & \frac{\partial e_{1,1}}{dw_N} \\ \frac{\partial e_{1,2}}{dw_1} & \frac{\partial e_{1,2}}{dw_2} & \cdots & \frac{\partial e_{1,2}}{dw_N} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial e_{1,M}}{dw_1} & \frac{\partial e_{1,M}}{dw_2} & \cdots & \frac{\partial e_{1,M}}{dw_N} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial e_{P,1}}{dw_1} & \frac{\partial e_{P,1}}{dw_2} & \cdots & \frac{\partial e_{P,1}}{dw_N} \\ \frac{\partial e_{P,2}}{dw_1} & \frac{\partial e_{P,2}}{dw_2} & \cdots & \frac{\partial e_{P,2}}{dw_N} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial e_{P,M}}{dw_1} & \frac{\partial e_{P,M}}{dw_2} & \cdots & \frac{\partial e_{P,M}}{dw_N} \end{bmatrix} \tag{58}$$

$$e = \begin{bmatrix} e_{1,1} & e_{1,2} & \cdots & e_{1,M} & \cdots & e_{P,1} & e_{P,2} & \cdots & e_{P,M} \end{bmatrix}^T \tag{59}$$

This results in the Gauss-Newton learning algorithm to recursively update the weights of the network.

$$w_{i+1} = w_i - (J_i^T J_i)^{-1} J_i e_i \tag{60}$$

Where the EBP is stable it converges slowly and while the Gauss-Newton method converges rapidly, it is unstable. The training method used in this work is the Levenberg-Marquardt method which is a combination of the steepest descent algorithm and the Gauss-Newton algorithm. The Levenberg-Marquardt method combines the strengths and omits the weaknesses of the EBP and Gauss-Newton methodology. It converges rapidly and is stable [17]. A combination coefficient, μ, is introduced and the learning method becomes

$$w_{i+1} = w_i - (J_i^T J_i + \mu I)^{-1} J_i e_i \tag{61}$$

where *I* is the identity matrix. The combination coefficient is used to switch between the EBP and Gauss-Newton methods during training. As μ approaches zero and the Gauss-Newton method is used. When it is large, the EBP method is used.

The computer program MATLAB provides a NN toolbox for the creation and training of various types of NN. For this work, a nonlinear input-output autoregressive neural network (NARX) is created and trained with the Levenberg-Marquardt method. Ten nodes were used along with four delays in order to maximize network performance while balancing the required computational infrastructure based on the principle that the number of nodes and delays determines how well the neural network can approximate the system being modeled [10]. The network architecture is shown in Figure 8.

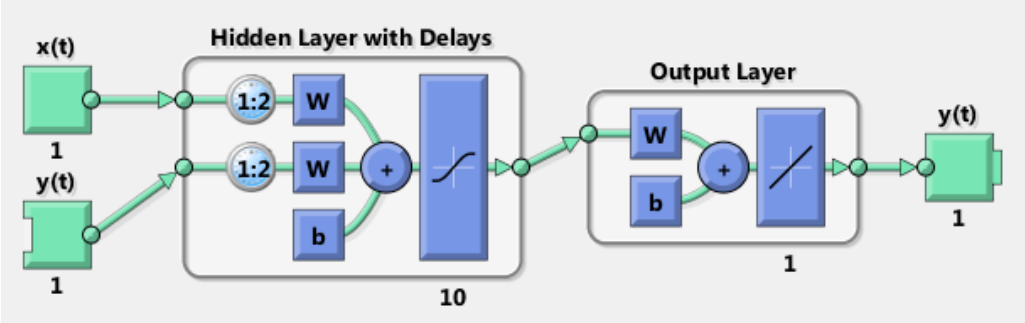

Figure 8. Network architecture of a NARX created in Matlab

Using this toolbox and the NARX network architecture , these methods can be applied to system identification (with the intent to learn complex input-output relationships). In terms of the regression model of Section IV.A known data, (e.g, the regressor data contained in φ), can be fed in along with known output data *y*. The network will then find an appropriate basis function and train the coefficients to closely match input to output. Another benefit to the NARX architecture over other architectures (including the simple non-linear input output model, or NIO) is that past values of *y* are fed back to the network if they are available to provide more accurate error estimation, online training, and an overall better estimate.

The major disadvantage of a NN model of a physical system is that the resulting model does not have any physical meaning. As discussed before, the end result of NN

estimation is a non-parametric (aka "black box") representation. The basis functions and weights assigned to them do not provide any physical insight into the system as Fossen's hydrodynamic coefficients do. So, while the resulting model may describe vehicle motion extremely well it cannot be used to infer other physical characteristics of the vehicle and must be validated by direct comparison of simulation to actual results. The value of the NN lies in its ability to learn complex, non-linear systems and account for these non-linearities through the training process whereas the parametric representation relies on assumptions and simplifications that do not always hold true. In the end, a very accurate model of a complex system can be produced with disproportionately small time and effort.

3. Gradient Estimator

The gradient estimator (GE) is the simplest of the various on-line estimation tools [2] and is a prediction-error-based method. The prediction error is defined as

$$e_i \triangleq \hat{y}_i - y_i \tag{62}$$

$$\tilde{\Theta} = \hat{\Theta}_i - \Theta \tag{63}$$

where $\hat{y}$ is the predicted output of Equation (47) at time step *i*, $\hat{\Theta}$ is a vector of the estimated parameters, and $\tilde{\Theta}$ is the parameter estimation error . The gradient estimator works by updating the parameters contained in Θ in the opposite direction of the gradient of Equation (62), the instantaneous prediction error, so that e_i is reduced.

$$\dot{\hat{\Theta}} = -p_0 \frac{\partial [e_l^T e_l]}{\partial \hat{\Theta}} \tag{64}$$

where p_0 is the estimator gain. Equation (64) can be re-written in terms of Θ and φ as

$$\dot{\hat{\Theta}} = -p_0 \varphi_i^T e_i \tag{65}$$

and in terms of the parameter estimation error $\tilde{\Theta}$ as

$$\dot{\tilde{\Theta}} = -p_0 \varphi^T \varphi \tilde{\Theta} \tag{66}$$

Slotine [2] proves the stability of Equation (65) using the Lyapunov candidate function

$$V = \Theta^T \Theta \tag{67}$$

and gives its derivative as

$$\dot{V} = -2 p_0 \tilde{\Theta}^T \varphi^T \varphi \tilde{\Theta} \leq 0 \tag{68}$$

which implies that the GE is stable and the magnitude of the squared parameter error is always decreasing. To guarantee that the GE will converge to the true value, however, the signal used to excite it must be sufficiently exciting. A signal is said to be sufficiently exciting if it possesses the appropriate level of persistence of excitation (PE). That is, the input signal must be sufficiently rich in frequency content (normally accomplished by using multiple sinusoids as inputs) to excite the relevant modes. Simply put, a signal composed of several sinusoids is capable of estimating more parameters than a signal of one sinusoid (this concept is explored in depth in Section IV.C). For this work, PE is not a limiting condition since multiple sinusoidal input signals may be applied to the vehicle to excite it.

For the gradient estimator PE is also essential to ensure the robustness of the estimator. Robustness is the ability of the estimator to maintain reasonably good parameter estimation in the presence of parametric uncertainties such as parameter time-variation, measurement noise, and disturbances. If the input signal is not sufficiently PE then the estimator values may diverge even without the presence of noise or uncertainty [2].

The quality of the estimates produced by the gradient estimator also depends on the rate of parameter variation, the level of non-parametric uncertainties, and the magnitude of the estimator gain [2].

The rate of variation of the parameters to be estimated affects the robustness of the estimator as well. If the parameters vary rapidly the estimator will have a much harder time accurately estimating the parameters and converging to the true value [2]. The hydrodynamic parameters to be estimated by this work are not expected to be time-varying so this limitation of the gradient estimator does not come into play here.[1]

[1] The parameters have been experimentally observed to vary with velocity. In order to compensate for this some time-variance may be assumed later. This assumption will be revisited in Chapter VI.

The presence of noise or un-modeled disturbances in the vehicle dynamics and measurements play a large part in the accuracy of the results produced by the GE. This is illustrated by adding a disturbance term, d_i, to y_i in Equation (62) resulting in

$$e_i = \hat{y}_i - (y_i + d_i) \tag{69}$$

Substituting Equation (69) into Equation (65) shows the effect of the disturbance

$$\dot{\hat{\Theta}} = -p_0 \varphi^T [\hat{y}_i - (y_i + d_i)] \tag{70}$$

which in turn affects parameter error through Equation (63). That is, for a large disturbance, the parameter estimation error becomes large. For this work, which involves noise in both the measurement noise from the motion capture system and process noise from the highly non-linear nature of a body moving in a non-viscous fluid, disturbances will greatly degrade the accuracy of a gradient estimator.

The effect of the magnitude of the estimator gain is easily seen in Equation (64). A larger gain results in a faster rate of convergence, but just as in optimization, increasing the gain or step size only improves performance up to a certain point. Increasing the gain to too high a level can result in oscillation and very slow convergence to the true parameters. Since the estimator gain is defined by the user this factor does not limit the use of a GE for this research.

4. Bayesian Filtering

Bayes' rule provides a method to compute a posterior probability from a set of given prior probabilities. It states that given a collection of k mutually exclusive and exhaustive events $A_1, A_2,\ldots,A_k$ that have their own prior probability $P(A_i)(i=1,\ldots,k)$, and any other event B, with $P(B) > 0$, the posterior probability of A_j given B has occurred, $P(A_j \mid B)$, can be expressed as [15]

$$P(A_j \mid B) = \frac{P(B \mid A_j)P(A_j)}{\sum_{i=1}^{k} P(B \mid A_j) \bullet P(A_j)} \quad \mathrm{j} = 1,\ldots,k \tag{71}$$

This concept can easily be applied to parameter estimation. Thrun *et al*. [3] present an adaptation that uses measurement and control input data to represent and evolve the belief state: the probability of state x_i given all the available information. The belief about a state, such as hydrodynamic parameters being estimated, at a time *i* is calculated from the belief at time *t*-1, the control input u_t at time *t*, and the measurement, y_i. For example, assuming that x_t has a Gaussian distribution, the belief state, *b*, can be described by the first and second moments of the distribution, mean, μ, and variance, σ^2.

$$b(x_i) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}} \tag{72}$$

Bayes rule provides an algorithm to recursively calculate the new belief state given the dynamic and measurement models, as well as the previous belief state and the measurement (e.g., [3]). It consists of two steps. The first step is called the prediction and it calculates the belief for state x_t, called $\bar{b}(x_t)$, based on the belief of the previous belief-state, $x_{t-1,}$ and the control input u_t.

$$\bar{b}(x_t) = \int_0^i p(x_i \mid x_{i-1}, u_i) b(x_{i-1}) dx_{i-1} \tag{73}$$

The second step is referred to as the measurement update. Here the Bayes filter uses the belief calculated by the prediction step in Equation (58) and multiplies it by the probability that measurement z_t was observed. Each hypothetical posterior state is treated this way and because the resulting product may not integrate to one the normalization factor η is used to ensure a proper probability is returned.

$$b(x_t) = \eta\, p(z_t \mid x_t) \bar{b}(x_t) \tag{74}$$

In order to use Equations (73)-(74) recursively $b(x_0)$ must be initialized at $i = 0$. Thrun *et al.* suggest that if x_0 is precisely known then $b(x_0)$ should be initialized as a point mass on the correct value of x_0 with zero probability elsewhere. If nothing is known about $x_{0,}$ however, a uniform distribution should be used for $b(x_0)$ [3]. Thrun *et al*. then gives the recursive form of Equations (73)-(74) as

$$\bar{b}(x_i) = \int p(x_i \mid x_{i-1}, u_t) b(x_{i-1}) dx_{i-1} \tag{75}$$

$$b(x_i) = \eta\, p(z_i \mid x_i) \bar{b}(x_i) \tag{76}$$

Practical implementation of Equations (75)-(76) requires the initial belief $b(x_0)$, the measurement probability $p(z_i \mid x_i)$, and the state transition probability $p(x_i \mid u_i, x_{i-1})$. The choice of the probability distributions for these required probabilities, as well as the system being modeled, shapes the type of Bayesian filter. For example, assuming a linear system, an initial Gaussian distribution, that all three probability distribution functions are zero mean, Gaussian white noise turns, and that the dynamic and measurement equations are linear the Bayesian filter into a Kalman filter

For the problem at hand, the system has non-linear dynamics described by:

$$\begin{aligned} x_{i+1} &= f(x_i, u_i, \omega_i) \\ \omega_i &\sim N(0, Q) \end{aligned} \tag{77}$$

where x_{i+1} is a vector of states at time *i+1*, x_i is a vector containing the states to be estimated at time step *i*, u_i is the control input at time step *i*, and ω_i is the zero mean, white noise with covariance *Q* associated with the process at time step *i*. The system has measurements z_k:

$$\begin{aligned} z_i &= h(x_i, \nu_i) \\ \nu_i &\sim N(0, R) \end{aligned} \tag{78}$$

where υ_k is the zero mean, white noise measurement noise with covariance *R* at time step *i*. One implementation of Bayes rule for non-linear systems is to linearize the system around trim points and to apply the Kalman filter machinery, resulting in different types of filters examples of which are the extended Kalman filter, unscented Kalman filter, and the particle filter [3]. To do this the concept of the Jacobian is used. The Jacobian is defined as the partial derivative of a process with respect to each of the variables that inform that process. Now, define the following *i* x *j* Jacobian matrices

$$
\begin{aligned}
A_{ij} &\triangleq \left.\frac{\partial f_i}{\partial x_j}\right|_{(\hat{x}_k, u_k, 0)} \\
W_{ij} &\triangleq \left.\frac{\partial f_i}{\partial \omega_j}\right|_{(\hat{x}_k, u_k, 0)} \\
H_{ij} &\triangleq \left.\frac{\partial h_i}{\partial x_j}\right|_{(\hat{x}_k, 0)} \\
V_{ij} &\triangleq \left.\frac{\partial h_i}{\partial \nu_j}\right|_{(\hat{x}_k, 0)}
\end{aligned}
\tag{79}
$$

Remembering that all noise distributions are assumed to be zero mean, Gaussian white noise and using an initial state, x_0, and an initial covariance matrix, P_0, the prediction step is:

$$
\begin{aligned}
\hat{x}_{i|i+1} &= f(\hat{x}_{i-1|i-1}, u_{i-1}, 0) \\
P_{i|i-1} &= A_{i-1} P_{i-1|i-1} A_{i-1}^T + W_{i-1} Q W_{i-1}^T
\end{aligned}
\tag{80}
$$

where $\hat{x}_{i|i-1}$ is the predicted state at time i given the state information from the previous step and $P_{i|i-1}$ is the predicted covariance at time i based on the covariance information from the previous step. Next is the measurement step

$$
\tilde{y}_i = z_i - h(\hat{x}_{i|i-1}) \tag{81}
$$

which consists of the measurement z_i and gives as its output the updated measurement error $\tilde{y}_i$. The Kalman gain, K_i, is then calculated as

$$
K_i = P_{i|i-1} H_i^T (H_i P_{i|i-1} H_i^T + V_i R V_i^T)^{-1} \tag{82}
$$

and the estimate $\hat{x}_i$ is updated by

$$
\hat{x}_{i|i} = \hat{x}_{i|i-1} + K_i (z_i - h(\hat{x}_{i|i-1}, 0)) \tag{83}
$$

Then the covariance matrix P_k is updated as

$$
P_k = (I - K_k H_k) P_{k|k-1} \tag{84}
$$

The process then repeats starting at Equation (80) allowing for recursive, on-line estimation of the parameters contained in $\hat{x}_i$.

B. PARAMETER ESTIMATION OF A DYNAMIC SYSTEM

The system identification techniques presented to date are developed for a static system of form $y_i = \varphi_i^1\theta^1 + \varphi_i^2\theta^2 + ... + \varphi_i^m\theta^m = \varphi_i^T\theta$. When identifying parameters for a dynamic system some additional steps are required to write the system in the form of (46) [11]. Consider the dynamic system:

$$\begin{aligned} \dot{x} &= f_0(x,u) + f(x,u) \\ \mathrm{y} &= Cx \end{aligned} \tag{85}$$

Θ is a matrix of unknown parameters and Φ*(x,u)* is a known regressor that consists of known basis functions. This dynamic model must be converted to a static system in order to apply the system identification techniques of the previous section. This can be accomplished through the introduction of filtered signals. First, rewrite Equation (85) as:

$$\dot{x} + ax = ax + f_0(x,u) + \Phi(x,u)\Theta \tag{86}$$

A filtered version of *x*, x_f, is also introduced as

$$\dot{x}_f + ax_f = ax \tag{87}$$

where a is the filtering time constant. Next, define $z = x - x_f$, then $\dot{z} = \dot{x} - \dot{x}_f$, allowing eq. (86) to be rewritten as:

$$\dot{z} + az = f_0(x,u) + \Phi(x,u)\Theta \tag{88}$$

The right hand terms can be thought of as forcing functions, and the first order ordinary differential equation has a known, unique solution:

$$z = e^{-at}z(0) + \int_0^t e^{-a(t-\tau)} f_0(x,u)d\tau + \int_0^t e^{-a(t-\tau)}\Phi(x,u)d\tau\Theta \tag{89}$$

Assume the initial conditions $x(0)=x_f(0)$ so that $z(0)=0$. Then equation (88) can be simplified to

$$z(t) = \Phi_0 + \Phi_f\Theta \tag{90}$$

where $\Phi_0 = \int_0^t e^{-a(t-\tau)} f_0(x,u)d\tau$ is the filtered version of $f_0(x,u)$ and $\Phi_f \triangleq \int_0^t e^{-a(t-\tau)}\Phi(x,u)d\tau$ is the filtered version of $\Phi_f(x,u)$. Furthermore, rewriting the measurement equation in terms of z and x_f, the dynamic system is converted into a static system:

$$y - Cx_f - C\Phi_0 = C\Phi_f\Theta \tag{91}$$

Since Φ_0 and Φ_f are filtered states, they can be calculated recursively as:

$$\begin{aligned}\dot{\Phi}_0 &= -a\Phi_0 + f_0(x,u)\\ \dot{\Phi}_f &= -a\Phi_f + f(x,u)\end{aligned} \tag{92}$$

Now, the dynamic system is presented as an equivalently static system consisting of filtered version of the known and unknown dynamics that make up the system. Because of this, the previously derived system identification techniques can be applied to the equivalent static system.

C. PERSISTENCE OF EXCITATION

The quality, or richness, of the signal used to excite a system during parameter estimation is of great importance for the quality of the estimate. For the GE it is required to ensure convergence to the true values of the parameters being estimated. Recall the equation for parameter estimation error of a GE, Equation (66). Solving this differential equation gives

$$\tilde{\Theta}_i = \tilde{\Theta}_0 \exp(-\int_0^i p_0\varphi^T(r)\varphi(r)dr) \tag{93}$$

where r is a dummy variable used for integration across the input signal $\varphi(r)$. This implies that

$$\lim_{i\to\infty}\int_0^i \varphi^T(r)\varphi(r)dr = \infty \tag{94}$$

Therefore, $\tilde{\Theta}_i$ will converge to zero (and therefore the estimated parameters will converge to the true values) if there are positive constants T and α such that for all $i > 0$

$$\int_{0}^{i+T} \varphi^T(r)\varphi(r)dr > \alpha I \tag{95}$$

where I is the identity matrix.

Recall from Equations (51) that to guarantee a unique solution to the RLLS problem $\Phi^T\Phi$ must not be singular. This concept is similarly discussed in depth by Astrom in [8] as requiring the matrix:

$$\Phi^T\Phi = \begin{bmatrix} \sum_{n+1}^{k} u^2(i-1) & \sum_{n+1}^{k} u(i-1)u(k-2) & \dots \sum_{n+1}^{k} u(i-1)u(k-n) \\ \sum_{n+1}^{k} u(i-1)u(k-2) & \sum_{n+1}^{k} u^2(k-2) & \dots \sum_{n+1}^{k} u(i-2)u(k-n) \\ \vdots & & \vdots \\ \sum_{n+1}^{k} u(i-1)u(k-n) & & \sum_{n+1}^{k} u^2(k-n) \end{bmatrix} \tag{96}$$

to be of full rank where u denotes the input signal, i is the time step, k is the total number of time steps, and n denotes the order of the system. Astrom defines this as the excitation condition. For long data sets all sums in Equation (96) are taken from 1 to k giving:

$$C_n \triangleq \lim_{k\to\infty} \frac{1}{k}\Phi^T\Phi = \begin{bmatrix} c(0) & c(1) & \dots & c(n-1) \\ c(1) & c(0) & \dots & c(n-2) \\ \vdots & & & \\ c(n-1) & c(n-2) & & c(0) \end{bmatrix} \tag{97}$$

where *c(i)* are the empirically determined covariances of the input signal such that

$$c(i) = \lim_{k\to\infty} \frac{1}{k} \sum_{i=1}^{k} u(i)u(i-k) \tag{98}$$

Interpreting Equations (96)–(98) helps define a signal as PE of order n if the limit of (97) exists and C_n is positive definite. It is important to recall here that the only guarantee required for existence of a unique solution to the RLLS problem is to satisfy that Equation (51) is full rank which is primarily driven by the regressor structure in Φ.

Therefore, if a non-PE signal is applied to a system the RLLS estimated values will converge to the true values, but only slowly Therefore, if it is possible, a PE signal should always be used to guarantee rapid convergence.

THIS PAGE INTENTIONALLY LEFT BLANK

V. SYSTEM IDENTIFICATION APPLIED TO VLBV300

The generic hydrodynamic model for an underwater vehicle was specialized for the vLBV300 by introducing various assumptions in Section III.B. Even with these simplifying assumptions (often times ignoring off-diagonal coefficients), the resulting model is highly coupled (see Equation (46) and 15 distinct parameters to be estimated). By exciting specific, uncoupled modes for the vLBV300 sequentially, individual parameters can be isolated and thus estimated. For example, by commanding thrust in the surge direction through X_{prop}, only motion in the surge direction is induced, reducing Equation (46) to:

$$\begin{aligned} \dot{u} &= \frac{X_{u|u|} u|u| + X_{prop}}{m - X_{\dot{u}}} \\ \dot{v} = 0, \dot{w} = 0, \dot{p} = 0, \dot{q} = 0, \dot{r} = 0 \end{aligned} \tag{99}$$

The filtering techniques of the previous chapter can then be applied to this simplified system to obtain a static system and to perform the system identification:

$$\begin{aligned} y_i &= \theta \varphi_i \\ \theta &= \begin{bmatrix} \frac{X_{u|u|}}{m - X_{\dot{u}}} & \frac{1}{m - X_{\dot{u}}} \end{bmatrix} \\ \varphi_i &= \begin{bmatrix} u|u| \\ X_{prop} \end{bmatrix} \end{aligned} \tag{100}$$

The inertia parameters are not identified specifically since they do not appear in the dynamic equations without their corresponding added mass term (e.g., I_X and $K_{\dot{p}}$ always appear together). This does not affect the applicability of the mode and the assumed values presented in Chapter III are used for clarification.

Since the SeaBotix ROV is also controllable in sway, heave, roll and yaw, the following simplifications are similarly possible and the system identification of the previous section can similarly be applied.

Sway:

$$\dot{v} = \frac{Y_{v|v|} v|v| + Y_{prop}}{m - Y_{\dot{v}}} \qquad (101)$$
$$\dot{u} = 0, \dot{w} = 0, \dot{p} = 0, \dot{q} = 0, \dot{r} = 0$$

Heave:

$$\dot{w} = \frac{Z_{w|w|} w|w| + Z_{prop}}{m - Z_{\dot{w}}} \qquad (102)$$
$$\dot{u} = 0, \dot{v} = 0, \dot{p} = 0, \dot{q} = 0, \dot{r} = 0$$

Roll:

$$\dot{p} = \frac{K_{p|p|} p|p| + F_{BK} + K_{prop}}{I_X - K_{\dot{p}}} \qquad (103)$$
$$\dot{u} = 0, \dot{v} = 0, \dot{w} = 0, \dot{q} = 0, \dot{r} = 0$$

Yaw:

$$\dot{r} = \frac{N_{r|r|} r|r| + N_{prop}}{I_Z - N_{\dot{r}}} \qquad (104)$$
$$\dot{u} = 0, \dot{v} = 0, \dot{w} = 0, \dot{p} = 0, \dot{q} = 0$$

However, control in the pitch degree of freedom is not possible (M_{prop}=0), but this mode can be excited by taking advantage of the coupling and commanding a combination of other modes. An examination of the dynamic equation associated with pitch reveals that by exciting the vehicle in the surge and heave degrees of freedom, the pitch portion of Equation (31) can be excited, resulting in the simplified to $\dot{v} = 0, \dot{p} = 0, \dot{r} = 0$ and:

$$\dot{u} = \frac{-(m - Z_{\dot{w}}) wq + X_{u|u|} u|u| + X_{prop}}{m - X_{\dot{u}}}$$
$$\dot{w} = \frac{(m - X_{\dot{u}}) uq + Z_{w|w|} w|w| + Z_{prop}}{m - Z_{\dot{w}}} \qquad (105)$$
$$\dot{q} = \frac{-(Z_{\dot{w}} - X_{\dot{u}}) uw + M_{q|q|} q|q| + F_{BM}}{(I_Y - M_{\dot{q}})}$$

In terms of Equation (85) define:

$$f_0 = \begin{bmatrix} \dfrac{-(m - Z_{\dot{w}})wq + X_{u|u|}u|u| + X_{prop}}{m - X_{\dot{u}}} \\ \dfrac{(m - X_{\dot{u}})uq + Z_{w|w|}w|w| + Z_{prop}}{m - Z_{\dot{w}}} \\ 0 \end{bmatrix} \tag{106}$$

$$\Phi = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ q|q| & F_{BM} - (Z_{\dot{w}} - X_{\dot{u}})uw \end{bmatrix} \tag{107}$$

$$\Theta = \begin{bmatrix} \dfrac{M_{q|q|}}{(I_Y - M_{\dot{q}})} & \dfrac{1}{(I_Y - M_{\dot{q}})} \end{bmatrix} \tag{108}$$

In an effort to verify the system identification approach and investigate persistence of excitation for the system, a simulator is developed according to the dynamics presented in Equation (46). Chen *et al*. [5] presents a model for the Seamor hovering-class ROV (the parameters were estimated using the Projective Mapping Method, while the mass and inertia parameters are listed as: mass = 20.4 kg, I_x = 0.429 kgm^2, and $I_y = I_z$ = 0.609 kgm^2). The results described in [5] are used as true values and the four system identification techniques described in Chapter IV are in turn applied to the simulated system to verify that the true parameters are estimated.

A. RLLS ESTIMATION

1. RLLS Estimation Applied to Simulator

The hydrodynamic coefficients are time-invariant in the simulator and a certain amount of measurement and process noise is expected. Recall from Equation (51) that $\Phi^T\Phi$ must be of full rank to guarantee the RLLS filter will converge to the true values of the parameters being estimated. As an example, the surge direction Equation (99) is investigated in this context. Here $\varphi_i = \begin{bmatrix} u|u| & X_{prop} \end{bmatrix}$ so

$$\Phi^T\Phi = \sum_{i=1}^{k} \begin{bmatrix} (u_i|u_i|)^2 & u_i|u_i|X_{prop,i} \\ X_{prop,i}u_i|u_i| & X_{prop,i}^2 \end{bmatrix} \tag{109}$$

The rank of the matrix described by Equation (109) was analyzed using data from the previously developed simulator and determined to be 2, or full rank. Therefore, according to the construction of the RLLS filter for this system the estimated parameter values will converge to the true values, given enough data (and time). This result is typical for the other DOF equations as well.

The RLLS estimator is applied to the simulator (with coefficients presented in [5]) to sequentially identify the hydrodynamic coefficients and verify the correct implementation of the RLLS method. As an example, results of the simulator RLLS surge testing are presented in Figure 13.

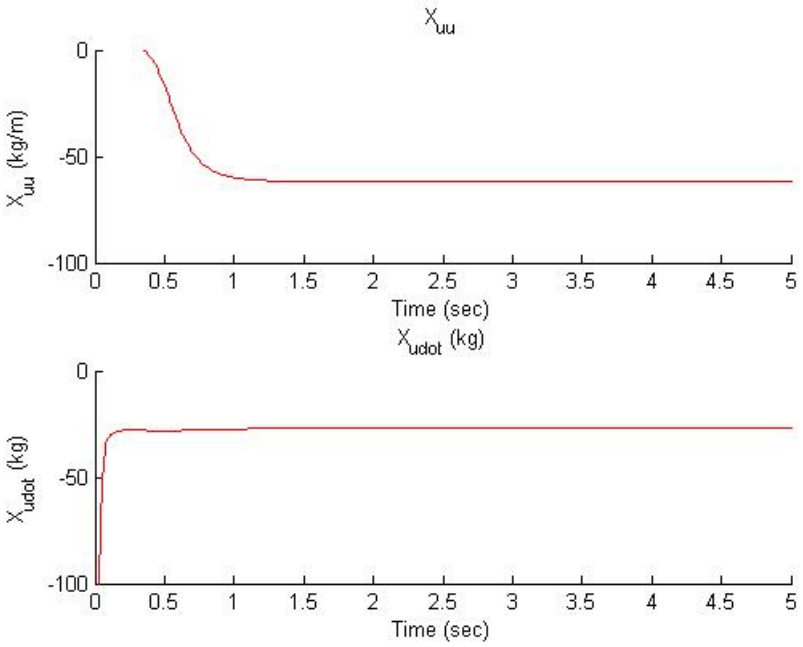

Figure 9. Convergence of simulator surge parameters

As discussed in Section IV.A and in [8], persistence of excitation is a construct for guaranteeing speed of convergence of the solution to the RLLS problem. The estimated parameter values can be seen to converge to the true values rapidly, implying that the system is persistently excited with a step function. A “stair-step” series of step inputs are more suitable if, as in this case, testing space and time are limited (due to the test tank facility).

These results for the surge direction are typical for the other coefficients with the exception of pitch. This is due to un-modeled dynamics represented by F_{BM} in Equation (37). Chen *et al*. [5] did not present the value for z_B, the position of the center of buoyancy relative to the center of gravity. Without knowing this value, assumed to be zero in the simulation, exact results cannot be achieved. The restorative force due to z_B shows up in the dynamics for the roll DOF as well but the contribution of z_B to F_{BK} is small due to small induced pitch angles and as such the parameters were still estimated accurately. Full estimation results are presented in Table 3.

	True value	Est. value	Units
$X_{\dot{u}}$	-27.08	-27.08	kg
$X_{u\|u\|}$	-61.117	-61.117	kg/m
$Y_{\dot{v}}$	-25.952	-25.952	kg
$Y_{v\|v\|}$	-139.81	-139.81	kg/m
$Z_{\dot{w}}$	-68.576	-68.576	kg
$Z_{w\|w\|}$	-51.724	-51.724	kg/m
$K_{\dot{p}}$	-61.683	-61.683	$kgm^{2/}/rad$
$K_{p\|p\|}$	-12.0	-12.0	$kgm^{2/}/rad^2$
$M_{\dot{q}}$	-79.411	-82.363	$kgm^{2/}/rad$
$M_{q\|q\|}$	-56.61	-58.77	$kgm^{2/}/rad^2$
$N_{\dot{r}}$	-0.154	-0.154	$kgm^{2/}/rad$
$N_{r\|r\|}$	-1.772	-1.772	$kgm^{2/}/rad^2$

Table 3. Comparison of hydrodynamic parameters

From these results, it can be concluded that the RLLS estimator performs properly and that the signal used to excite the system contains sufficient PE.

2. RLLS Applied to SeaBotix vLBV300

Since the vLBV300 is not yet equipped with an onboard Inertial Navigation System (INS), an external motion capture system (VICON) is used to measure the vehicle

position and orientation in the NPS CAVR test tank. From these pose measurements, the linear and angular velocities are estimated. The VICON system consists of Infrared LED arrays and cameras that track reflective markers in the operating space. Due to the absorption of electromagnetic signals in water, the current system is only applicable for in-air operation. To overcome this limitation, the vLBV300 has been extended above the water surface with a light-weight, low inertia structure. By tracking this structure and performing the appropriate coordinate transformations, the submerged vehicle's motion can be tracked. This structure has a small effect on the dynamics of the vehicle, and a more appropriate solution is being investigated (such as relying on INS data instead). The VICON system provides high-accuracy data (<1cm) at high data rates (100 Hz).

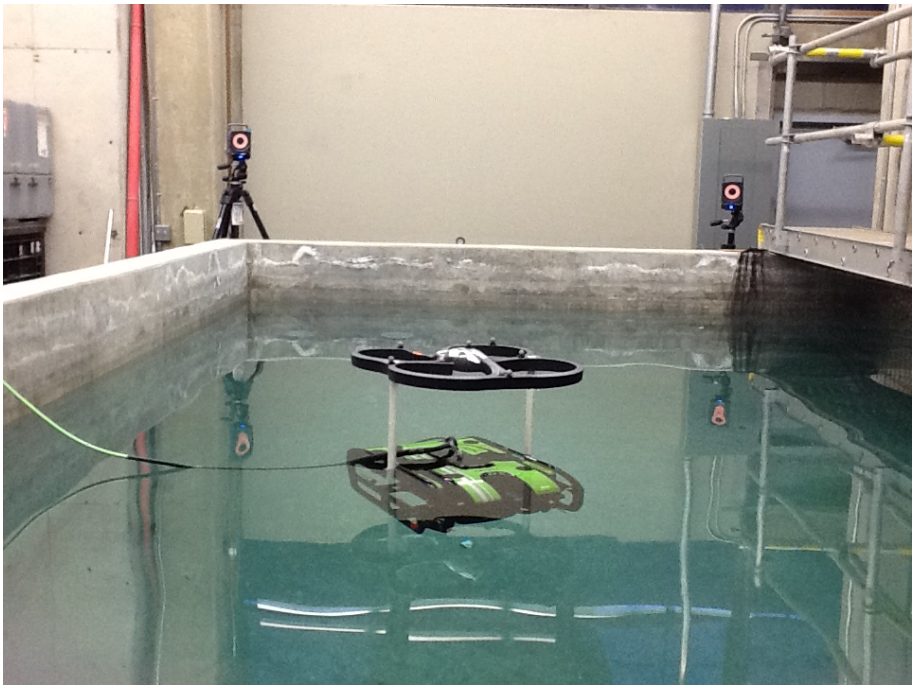

Figure 10. SeaBotix vLBV300 in the instrumented NPS dive tank

There are some limitations associated with this experimental setup, the primary of which is the size of the NPS dive tank. At approximately 1.4 meters deep, 4.5 meters wide and 6.5 meters long the size of the tank prohibits extended data collection, an effect that is exacerbated during higher speed runs. In order for the VICON system to provide, data it must be able to see the motion capture "pucs" (IR reflectors) on top of the

vehicle's extension. As the vehicle approaches the extremes of the tank, these reflectors may become occluded from the cameras, causing jumps in the data and further limiting the space available for experiments. Another limitation is the VICON system itself. While VICON provides very high accuracy in position and pose data it is a physical system and is therefore subjected to some measurement noise. VICON does not provide velocities or angular rates and therefore the provided position data must be filtered and differentiated (and transformed as required) to obtain the required body velocities and angular rates. This differentiation has the effect of magnifying the noise present in the VICON measurements. A low-pass filter is used to remove high frequency noise in the measurements but those effects are still apparent in the output velocities and angular rates. In the near future, CAVR will be installing an INS into the SeaBotix vLBV300. This will help to improve the parameter estimation results by removing the need to externally measure and process position data as well as allowing sustained, at-sea data runs. All of these effects are seen in the application of RLLS to the SeaBotix vLBV300 but do not prevent system identification results from being obtained.

One of the fundamental assumptions for the generic hydrodynamic coefficients for the system introduced in Chapter III is time and state independence. This may be a valid assumption for systems operating around a trim point, as is implicitly the case for the simulator (and thus the results for the previous section). However, in practice these coefficients are dependent on the system state, in particular vehicle velocity. This can be seen in the initial application of the RLLS estimator to the vLBV300 using real-world data. The RLLS estimator is used to estimate the surge parameters of the vLBV300 during a run with a PWM command of 20 to the aft thrusters only. The resulting coefficients are used to obtain a response which is compared to a second experimental run also conducted at a PWM of 20 to the aft thrusters. The surge displacement results are presented in Figure 11 and the velocity results are presented in Figure 12.

The simulated results did not match the experimental results as expected, and in fact differed by a significant amount. The discrepancy derives from the speeds at which the coefficients were derived and the speed at which the vehicle moved in the experimental data. The average steady state surge velocity u of the run used to determine

the coefficients was 0.3635 m/s while the value of *u* for the experimental run used for comparison was 0.2915 m/s. The effects of this are clearly seen in Figures 11 and 12.

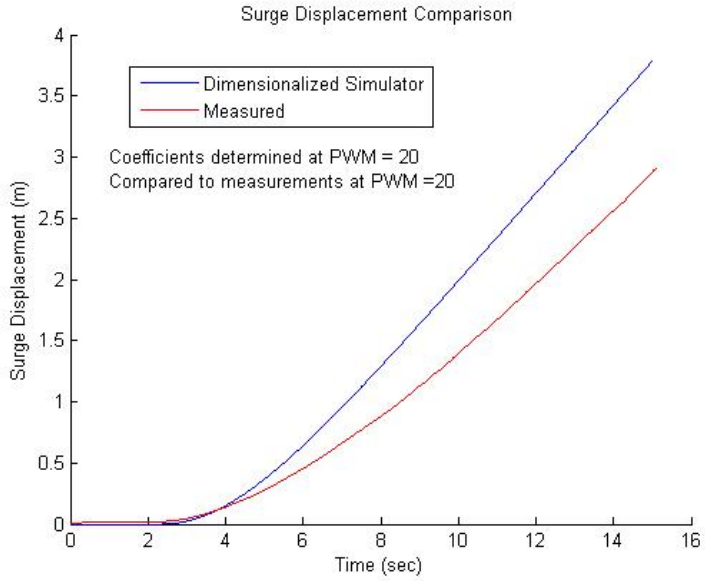

Figure 11. Simulated surge displacement compared to experimental surge displacement at a PWM command of 20

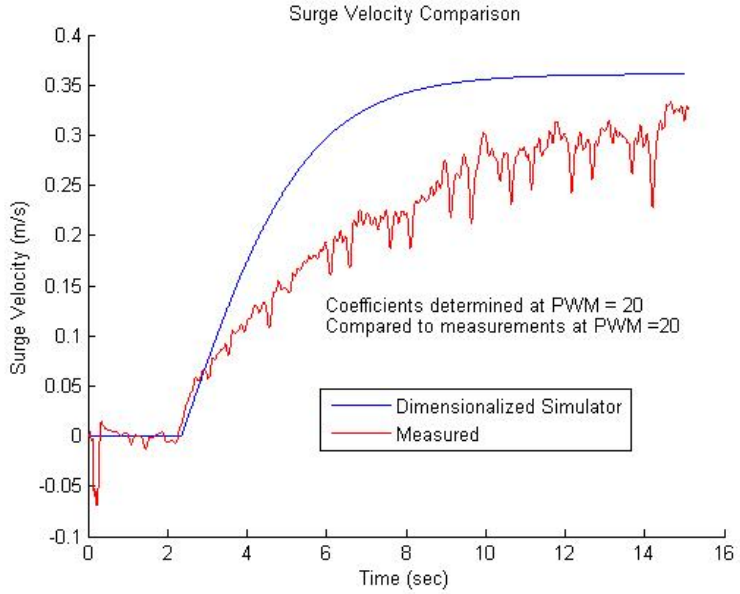

Figure 12. Simulated surge velocity compared to experimental surge velocity at a PWM command of 20

One approach to capture the dependence of these coefficients on velocity in particular is to introduce dimensionless coefficients. Fossen [1] presents a table of normalization variables (the Prime I system), which is used to achieve the desired non-dimensional form. The normalization variables relevant to this work are presented in Table 4.

Units	Prime-system I
Mass	$\frac{\rho}{2}L^3$
Inertia	$\frac{\rho}{2}L^5$
Force	$\frac{\rho}{2}U^2L^2$
Moment	$\frac{\rho}{2}U^2L^3$
Quadratic Damping	$\frac{\rho}{2}L^2$
Linear Damping	$\frac{\rho}{2}UL^2$

Table 4. Normalization variables used in the Prime I system (From [1])

L is the characteristic length of the vehicle which defines the scale of the vehicle. In this case it is chosen to be the length of the vehicle along its x axis, or 0.625 meters. ρ is the density of water at the temperature of the tank in which the vehicle is being tested. For this work that is assumed to be the density of freshwater at 15 degrees centigrade, $\rho = 1,000\frac{kg}{m^3}$. Finally, let the steady state velocity of the vehicle be

$$U = \sqrt{u^2 + v^2 + w^2} \tag{110}$$

In the following, the subscript c applied to U denotes the value used to derive the coefficients while the subscript r applied to U denotes the value used during the comparison run.

Using the variables from Table 4, it is possible to define a prime set of dimensionless variables. As an example, the variables from Equation (99) for surge are shown in detail here while only the results of the other two planar DOF are presented.

$$m' = \frac{m}{\frac{\rho}{2}L^3} \tag{111}$$

$$X_{\dot{u}}' = \frac{X_{\dot{u}}}{\frac{\rho}{2}L^3} \tag{112}$$

$$X_{u|u|}' = \frac{X_{u|u|}}{\frac{\rho}{2}L^2} \tag{113}$$

$$X_{prop}' = \frac{X_{prop}}{\frac{\rho}{2}U_c^2 L^2} \tag{114}$$

Inserting Equations (111)-(114) into Equation (99) gives the non-dimensional equation

$$\dot{u} = \frac{X_{u|u|}'}{L(m' - X_{\dot{u}}')} u|u| + \frac{U_r^2}{L(m' - X_{\dot{u}}')} X_{prop}' \tag{115}$$

Similarly, then

$$\dot{y} = \frac{Y_{v|v|}'}{L(m' - Y_{\dot{v}}')} v|v| + \frac{U_r^2}{L(m' - Y_{\dot{v}}')} Y_{prop}' \tag{116}$$

$$\dot{w} = \frac{Z_{w|w|}'}{L(m' - Z_{\dot{w}}')} w|w| + \frac{U_r^2}{L(m' - Z_{\dot{w}}')} Z_{prop}' \tag{117}$$

Since Equation (104) involves an inertia term, I_z, instead of a mass term as before, the coefficients become

$$I_Z' = \frac{I_Z}{\frac{\rho}{2}L^5} \tag{118}$$

$$N_{\dot{r}}' = \frac{N_{\dot{r}}}{\frac{\rho}{2}L^5} \tag{119}$$

$$N_{r|r|}' = \frac{N_{r|r|}}{\frac{\rho}{2}L^3} \tag{120}$$

$$N_{prop}' = \frac{N_{prop}}{\frac{\rho}{2}U_c^2 L^3} \tag{121}$$

Inserting Equations (118)-(121) into Equation (104) gives the non-dimensional equation

$$\dot{r} = \frac{N_{r|r|}'}{L^2(I_Z' - N_{\dot{r}}')} r|r| + \frac{U_r^2}{L^2(I_Z' - N_{\dot{r}}')} N_{prop}' \tag{122}$$

It is easy to see that Equations (115)-(117) and (122) can still be re-arranged to fit the form of Equation (47) so the RLLS may still be used to estimate the unknown parameters. To fully investigate the vehicle and verify the normalization scheme used, two surge runs are presented at high speed (PWM command of 50) and low speed (PWM command of 20) to the aft thruster pair. Dimensionalized and non-dimensionalized coefficients are determined for both runs and initially compared to different runs at the same PWM command that the coefficients are estimated at. Slightly different steady state velocities are determined for the two runs at the same PWM command for both the high and low speed runs. The high speed run results are presented in Figure 13 and the slow speed results are presented in Figure 14.

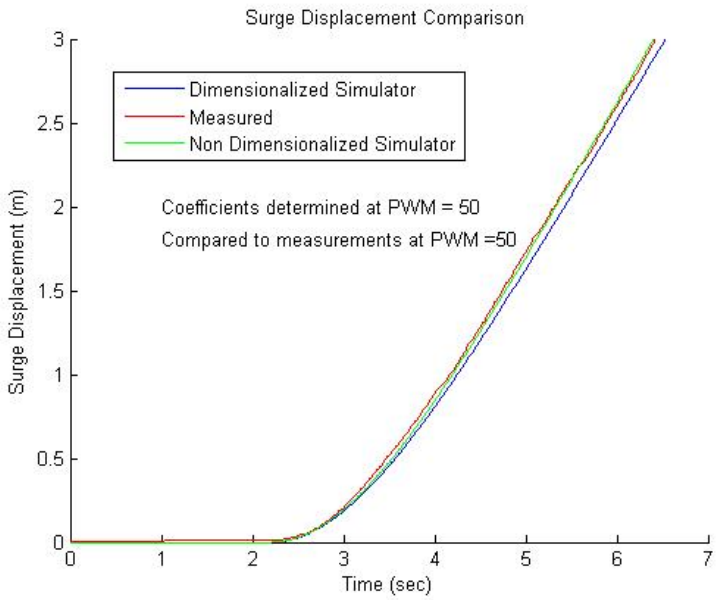

Figure 13. Simulated surge displacement results compared to measured surge displacement results at a PWM command of 50 using coefficients determined at a PWM command of 50

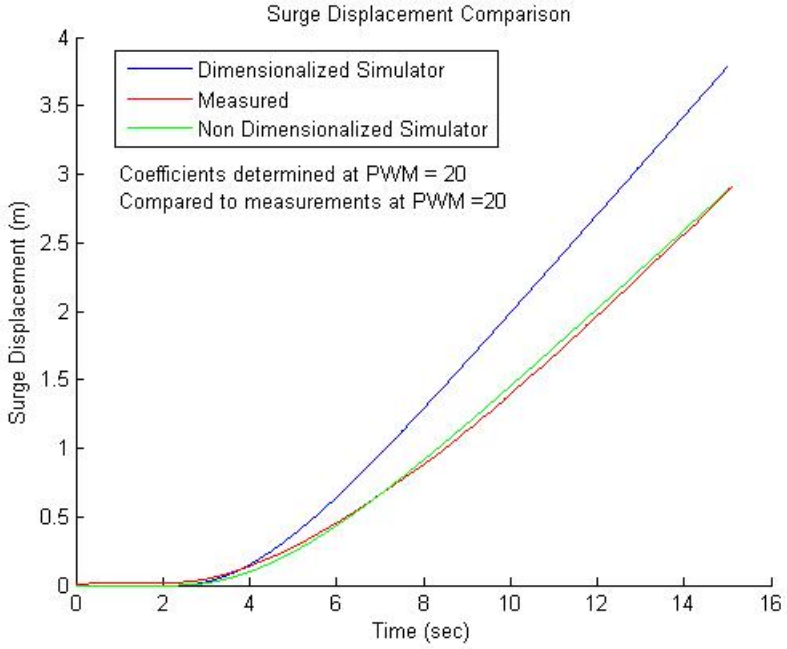

Figure 14. Simulated surge displacement results compared to measured surge displacement results at a PWM command of 20 using coefficients determined at a PWM command of 20

In both cases it can be seen that the non-dimensional simulator performs more accurately than the dimensionalized version, but in the high speed run the dimensionalized simulator also performs well. Additionally, when coefficients derived during the low speed run are used to simulate a high speed run (or vice versa) the results diverge significantly from the experimental data. This is in part due to the application of the model away from the trim condition. Figure 15 shows the divergence of the high speed coefficients used to simulate a low speed run. Low speed coefficients simulating a high speed result also demonstrate this effect and diverge significantly.

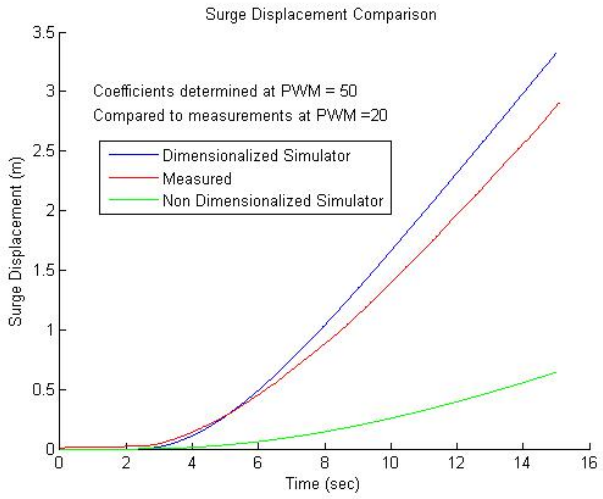

Figure 15. Comparison of surge displacement to experimental displacement using coefficients determined at PWM command of 50 and a simulated run at a PWM command of 20.

Both the inaccuracy in the dimensionalized simulator seen in Figures 13 and 14 and the divergence of the non-dimensionalized simulator shown in Figure 15 can be explained by an examination of the Reynolds number. The Reynolds number is a dimensionless parameter that relates the viscous behavior of all Newtonian fluids [20]. It is expressed mathematically as

$$\mathrm{Re} = \frac{\rho V L}{\mu} \tag{123}$$

where V and L are the characteristic velocity and length that describe the vehicle's passage through the fluid, ρ is the density of the fluid the vehicle is travelling in, and μ is the viscosity of the fluid. The Reynolds number is also used to determine if a condition of similarity exists between the results determined from two different experiments. White defines the similarity condition as existing if the Reynolds number of one run is equal the Reynolds number of the second run. He demonstrates this using a ratio of generic force coefficients [20]

$$\frac{F_1}{F_2} = \frac{\rho_1}{\rho_2}\left(\frac{V_1}{V_2}\right)^2 \frac{L_1}{L_2}^2 \tag{124}$$

According to Equation (124) a scaled relationship between the two force coefficients exists if the Reynolds number of the two experiments is the same. For this work, since the density of the water and the characteristic length of the vehicle is the same between experiments Equation (124) simplifies to

$$\frac{F_1}{F_2} = \left(\frac{V_1}{V_2}\right)^2 \tag{125}$$

which shows that the velocities must be equal in order for the condition of similarity to exist. Because the experiments are conducted at different velocities the Reynolds numbers are not the same and the coefficients cannot be scaled. This difference is not noticeable in Figures 13 and 14 because the steady state velocities are similar enough. (i.e., it is close enough to the trim condition). This is not the case for the high-vs low-speed runs.

The second conclusion drawn from an analysis of the Reynolds number explains the difference between the dimensionalized and non-dimensionalized simulators in the low speed analysis. Recall another assumption previously made that only turbulent flow would exist around the vehicle during its motion. This allowed for simplification of the damping matrix $D(\nu)$ to only include the quadratic damping terms and allowed for the neglect of the linear damping terms.

The Reynolds number is also a good indicator of which type of flow will be present around a body moving in a fluid. White states that turbulent flow will be present at Reynolds numbers greater than approximately 10^6 [20]. Calculating the Reynolds numbers for the low speed and high speed runs gives

$$\mathrm{Re}_{low} = \frac{1000\frac{kg}{m^3}0.3635\frac{m}{s}0.625kg}{1.1x10^{-3}Pa-s} = 2.06x10^5$$
$$\mathrm{Re}_{high} = \frac{1000\frac{kg}{m^3}0.8630\frac{m}{s}0.625kg}{1.1x10^{-3}Pa-s} = 4.903x10^5 \quad (126)$$

These Reynolds numbers show the flow around the vehicle should be totally laminar for the low speed run, not turbulent, and therefore that ignoring the linear damping terms is not a good assumption. This effect is particularly visible in the low speed run shown in Figure 14. The fact that the high speed run in Figure 13 matches well implies that for the vLBV300 turbulent flow is beginning to dominate around 0.9 meters per second therefore the vehicle is likely operating at the end of the transition region. The damping matrix of the model, then, must be updated to include the linear damping coefficients instead of quadratic damping for the low speed run.

$$D(\upsilon) = -\mathrm{diag}\left(\left[\, X_u, Y_v, Z_w, K_p, M_q, N_r \,\right]\right) \quad (127)$$

Equations (99)-(104) then become

$$\dot{u} = \frac{X_u u + X_{prop}}{m - X_{\dot{u}}}$$
$$\dot{v} = \frac{Y_v v + Y_{prop}}{m - Y_{\dot{v}}}$$
$$\dot{w} = \frac{Z_w w + Z_{prop}}{m - Z_{\dot{w}}} \quad (128)$$
$$\dot{p} = \frac{K_p p + F_{BK} + K_{prop}}{I_X - K_{\dot{p}}}$$
$$\dot{r} = \frac{N_r r + N_{prop}}{I_Z - N_{\dot{r}}}$$

and Equation (108) becomes

$$\dot{u} = \frac{-(m - Z_{\dot{w}})wq + X_u u + X_{prop}}{m - X_{\dot{u}}}$$
$$\dot{w} = \frac{(m - X_{\dot{u}})uq + Z_w w + Z_{prop}}{m - Z_{\dot{w}}} \tag{129}$$
$$\dot{q} = \frac{-(Z_{\dot{w}} - X_{\dot{u}})uw + M_q q + F_{BM}}{(I_Y - M_{\dot{q}})}$$

RLLS regression is again applied to the data sets surge, sway, heave, and yaw using the updated model regressor form

$$\dot{u}_i = \begin{bmatrix} \frac{X_u}{(m - X_{\dot{u}})} & \frac{1}{(m - X_{\dot{u}})} \end{bmatrix} \begin{bmatrix} u \\ X_{prop} \end{bmatrix} \tag{130}$$

as well as the updated non-dimensionalized regressor form. The surge equations are presented as an example because the other updated DOF equations are typical to surge.

$$y_i = \begin{bmatrix} \frac{X_u' U}{L(m' - X_{\dot{u}}')} & \frac{U^2}{L(m' - X_{\dot{u}}')} \end{bmatrix} \begin{bmatrix} u \\ X_{prop}' \end{bmatrix} \tag{131}$$

$$X_u' = \frac{X_u}{\frac{\rho}{2} L^2} \tag{132}$$

The surge displacement results using the linear damping matrix for a slow speed run are presented in Figure 16 and show that the damping model is still not entirely correct.

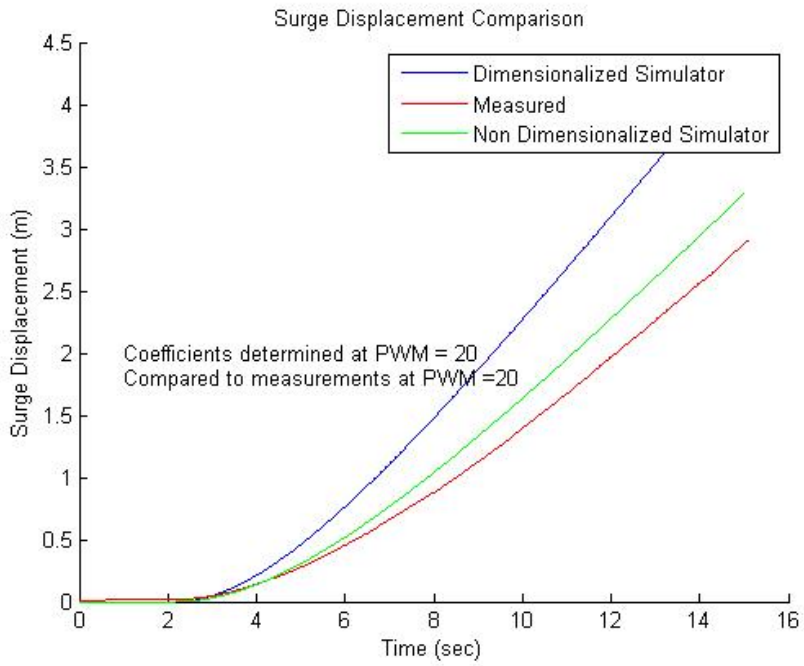

Figure 16. Surge displacement comparison using linear only damping model

Based on these results it is concluded that the true force due to damping is probably a combination between linear and quadratic, between laminar and turbulent. While the Reynolds number for this run suggests total laminar flow, the equation used for this is the Reynolds number of an infinitely flat, smooth plate. The SeaBotix vLBV300 is clearly not a smooth, flat plate. Thus, it is more likely that the vehicle operates in the transition region from laminar to turbulent flow. Most likely, there are local instances of turbulent flow due to the rough surface of the sides and underside of the vehicle while laminar flow exits on the largely flat upper surface. The damping matrix can be represented as a combination of both the linear and quadratic terms and this is a good area for further work.

Since the quadratic damping model performed well at a PWM command of 50 for both the dimensionalized and non-dimensionalized simulators the full results for the high speed experiments in surge, sway, heave, and yaw are presented here as a starting point for future work. As demonstrated, these values will only be accurate around the speed at which they are derived but they are useful as a starting point for future work.

Complete system identification results for the SeaBotix vLBV300 at high speeds are presented in Table 5. The high speed runs for surge and sway are conducted at a PWM command of 50 while the heave and yaw runs were conducted at PWM command of 40 (the shallowness of the tank in the heave DOF and the tether wrapping around the vehicle limited operations). The convergence of the surge coefficient values, both dimensionalized and non-dimensionalized, is presented in Figure 17 and the sway and yaw DOF results are similar to this. The heave DOF results are presented in Figure 18 to demonstrate the limitation imposed by the shallow depth of the tank. In order to prevent the vehicle from colliding with the bottom of the tank, only about two seconds of data is available to the estimator. Figure 18 shows that the RLLS appears to converge then makes a correction (both in the dimensional and non-dimensional filter) at the very end. Clearly, longer data runs would provide more time for the filter to converge and therefore more accurate results.

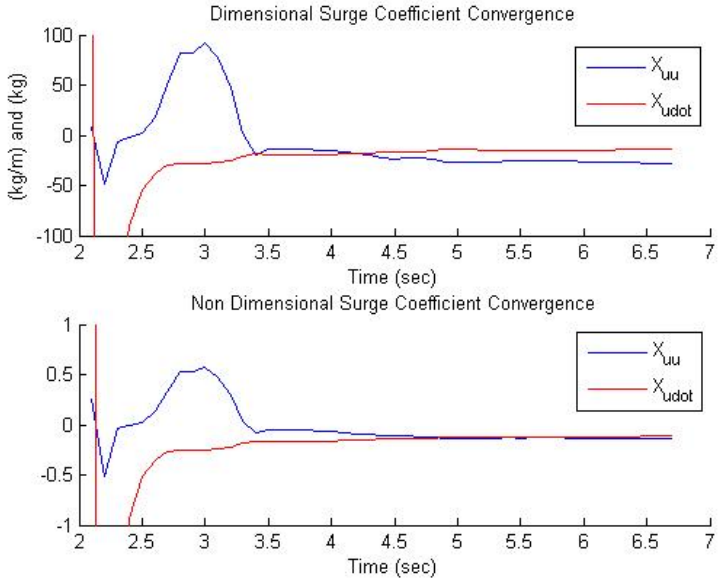

Figure 17. Convergence of surge coefficients at high speed

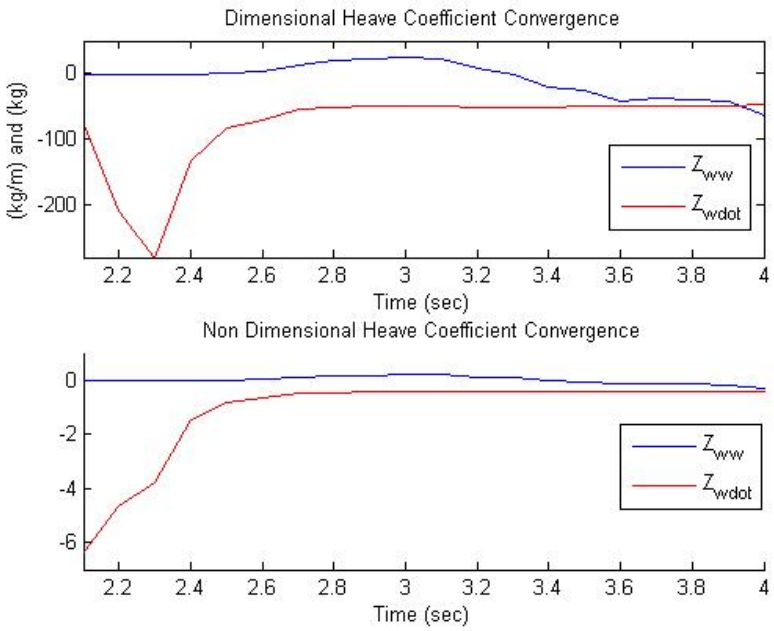

Figure 18. Convergence of heave coefficients at high speed

	Dimensionalized	Units	Non-Dimensionalized
$X_{\dot{u}}$	-13.5778	kg	-0.1146
$X_{u\|u\|}$	-27.7411	kg/m	-0.1416
$Y_{\dot{v}}$	-27.9347	kg	-0.2334
$Y_{v\|v\|}$	-50.6868	kg/m	-0.2592
$Z_{\dot{w}}$	-46.3258	kg	-0.3971
$Z_{w\|w\|}$	-64.5970	kg/m	-0.3000
$N_{\dot{r}}$	-3.1023	$kgm^{2/}/rad$	-0.8112
$N_{r\|r\|}$	-2.1709	$kgm^{2/}/rad^2$	-0.3416

Table 5. Parameter estimation results for SeaBotix vLBV300 at high speed

3. Conclusion: RLLS Estimator

The RLLS estimator is applied to both a dimensionalized and non-dimensionalized model for the SeaBotix vLBV300 platform. The benefits of the non-dimensionalized approach are demonstrated to make small adjustments around the trim

conditions. The non-dimensional RLLS produced both a low-speed and a high-speed model that performed well when tested at their respective speeds and generally performed better than the dimensionalized simulator. A Reynolds number analysis and supporting experimental data suggests that the vehicle operates in different regimes for the low-and high-speed operations and an alternative model for low-speed operations is investigated. From these results it is concluded that a combination of these models may be required. Also, since there is a strong dependence of the model coefficients on vehicle state (velocity in particular), it may be more appropriate to model the system with time-varying parameters, which can also be estimated with system identification techniques (recommended future work).

The RLLS is an excellent estimator for this work. It has the ability to estimate parameters of both constant and time-varying parameters, which as demonstrated, may be a useful extension of this work to provide the most accurate hydrodynamic parameter results possible. A major current limitation is the size of the NPS dive tank, which does not allow for data runs longer than approximately 12 to 15 seconds at low speed or five-to-seven second runs at high speed. In the presence of noise and uncertainty this is often not enough time for the filter to converge. Longer data runs would greatly benefit the RLLS by allowing it more time to converge.

Additionally, this portion of the work identified the ideal sequence of commanded, decoupled motion to identify the full set of model parameters. Because of the ability to perform uncoupled motion in the surge, sway, heave, roll, and yaw DOF the parameters of those equations should be determined in that order. This is required to allow for the identification of the pitch coefficients through coupling of surge and heave. After these primary terms have been determined, they can be plugged back into the appropriate equations of motions (used in subsequent estimation runs).

This strategy can also be applied through careful analysis of the equations of motion to identify more coupling terms (to relax some of the assumptions). The coupled motions can be performed in all possible permutations (surge-sway, heave-roll, sway-heave, etc.) in order to determine the cross-coupling terms that were initially neglected in the equations of motions. Finally, with these initial parameters in place, the filter can be

deployed on-line on the vehicle to constantly learn and update from these baseline values. As more data is collected through normal operations the estimates can be refined or can adapt to mid-mission configuration changes.

B. NEURAL NETWORK ESTIMATION

1. Neural Network Applied to Simulator

The strength of the neural network approach is that, using sufficient number of nodes, an arbitrary number of inputs can be mapped to an arbitrary number of outputs with good accuracy [10]. The type of NN investigated is the Non-linear Autoregressive (with External Input) NN (NARX). This type of model architecture is generally more accurate than other models because it incorporates feedback. That is, previous values of the output y are fed back to help improve the accuracy of the network and continually train the network. As an example, thruster force in the surge direction was mapped to surge velocity for the simulated Seamor ROV. Both the X_{prop} command and the resulting surge velocity are corrupted by low power, white noise. First, a network using three nodes and two delays is trained then a network using 10 nodes and two delays is trained The diagnostic graph of the NARX response for the three node network is presented in Figure 19 while the response for the 10 node network is presented in Figure 20.

In the learning block, 70 percent of the supplied data was used to create and train the network, 15 percent was used to validate the network, and the final 15 percent was used to further test the network. The top graph shows the response of the output element one, or for this network, surge velocity u, over time plotted against test and validation points. The lower graph shows the magnitude of the error between the network output and the supplied, true output data. .

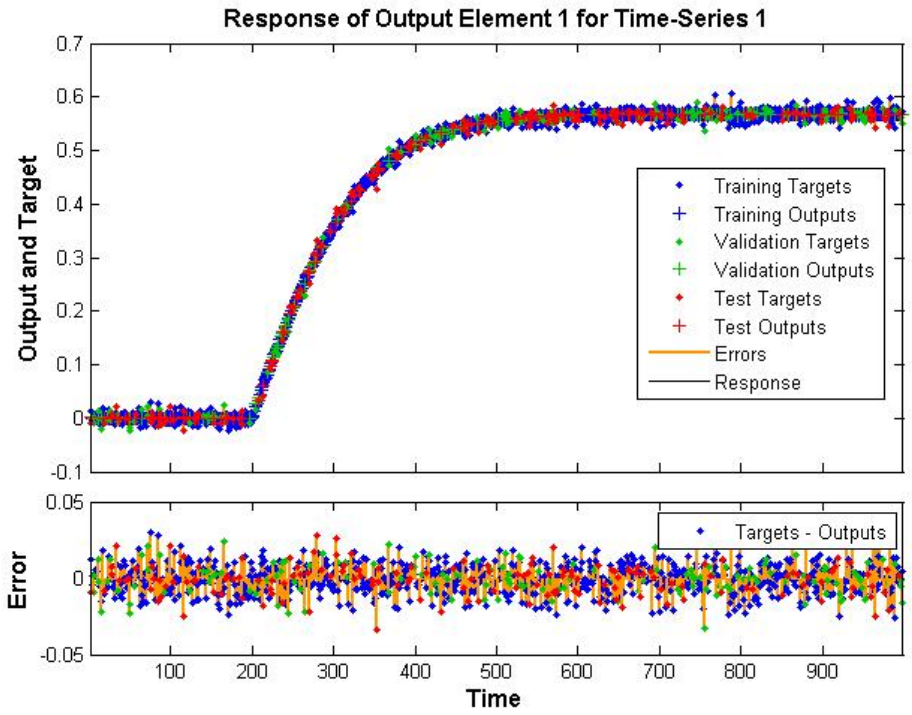

Figure 19. Diagnostic plot of NARX with three nodes and two time delays

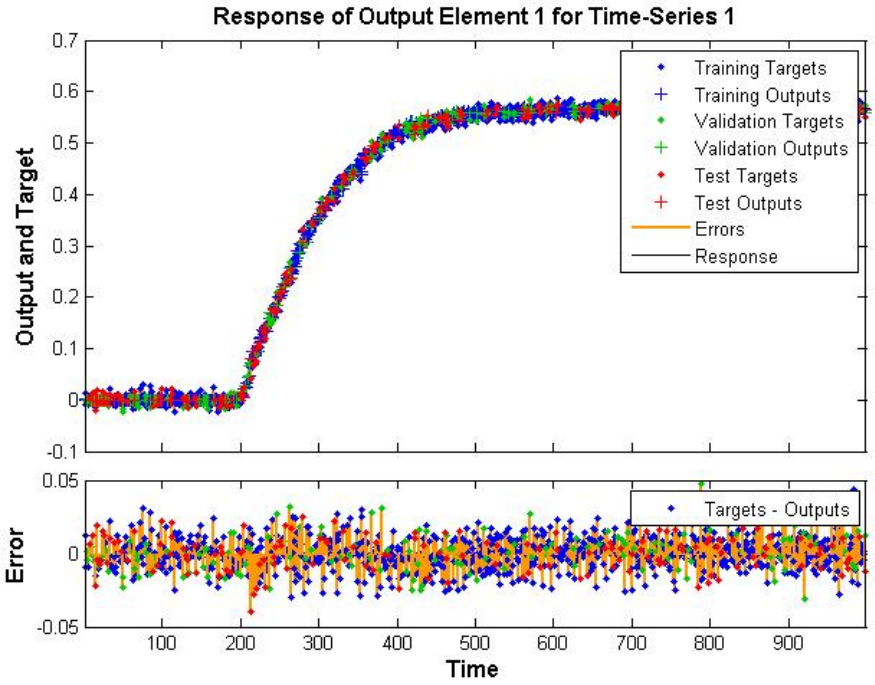

Figure 20. Diagnostic plot of NARX with 10 node and two time delays

A comparison between Figures 19 and 20 shows that increasing the number of nodes from three to 10 has little effect on the magnitude of the error. However, a comparison of their outputs when plotted against the simulated surge velocity demonstrates another constraint of NN: using too many nodes to model a simple system. This comparison of the surge velocities produced by both networks is presented in Figure 21.

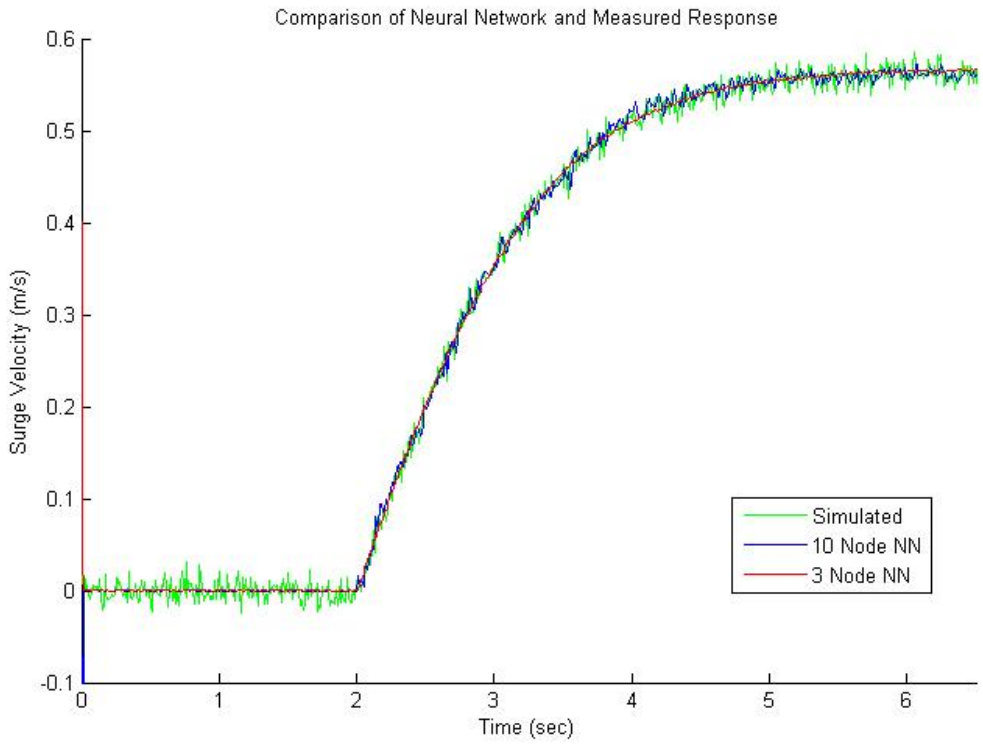

Figure 21. Comparison of actual surge velocities to 10 node and three node NARX

Figure 21 shows that, for the simple input-output relationship between X_{prop} and surge velocity a three node network is capable of modeling the response of this system. It also does a good job of rejecting the high frequency noise used to corrupt the inputs and outputs. The 10 node network, however, attempts to model the unwanted high frequency noise content. This demonstrates the power of increasing the size of a NN as well as the dangers of doing so. Therefore, when choosing a NN size careful

consideration must be given to using enough nodes to adequately model the system while ensuring the noise inherent to the system is not also modeled by the NN.

Another important consideration when modeling using NN is the data used to train the network. The network will only be as good as the information contained in the training data. For example, when initially training the NN used to map the X_{prop} to surge velocity, a disproportionate amount of the data used was for the steady state condition of the simulator after it reached cruising speed. Since the NN toolbox randomly chooses points from the data provided to train the network, this resulted in poor modeling of the transient. Truncating the data used to train the NN to involve an equal amount of data for the transient and steady state generated the much better results seen in Figure 21. Therefore, if a single NN is expected to model an entire complex system then when training it is crucial to do so values for all possible operational ranges and conditions and to ensure all dynamics are equally represented in the amount of data provided to train the system.

Next, a 10 node, four time delay NARX neural network is created and trained for all four controllable DOF of the Seamor ROV. This is accomplished by exciting the four controllable DOF sequentially in the order surge, sway, heave, yaw then exciting the vehicle in the coupled modes of surge-sway, surge-heave, surge-yaw, sway-heave, sway-yaw, heave-yaw, surge-sway-heave, and surge-sway-yaw. The goal of this input sequence is to excite the vehicle as completely in all six DOF (relying on coupling effects for roll and pitch) as possible in order to ensure the information used to train the NARX is a rich as possible. The output values used to train the NARX are the simulated body velocity and angular rate values *u, v, w,* and *r*. The input commands and output measurements are corrupted with low power, white noise to make the simulation as realistic as possible. The diagnostic results relevant to the surge output element are presented in Figure 22.

Because of the computational infrastructure available for this work, trying to include roll and pitch is not possible. Nor are longer data runs. The computer used cannot handle the training process for even a small, 10 node network using six inputs, six outputs, and the large array of training data. It is also difficult to increase the number of

nodes for non-simple (one input to one output) systems which again demonstrates the importance of choosing a network size to adequately model the system while avoiding "over-modeling" and over-burdening of the computer used to create and train the network.

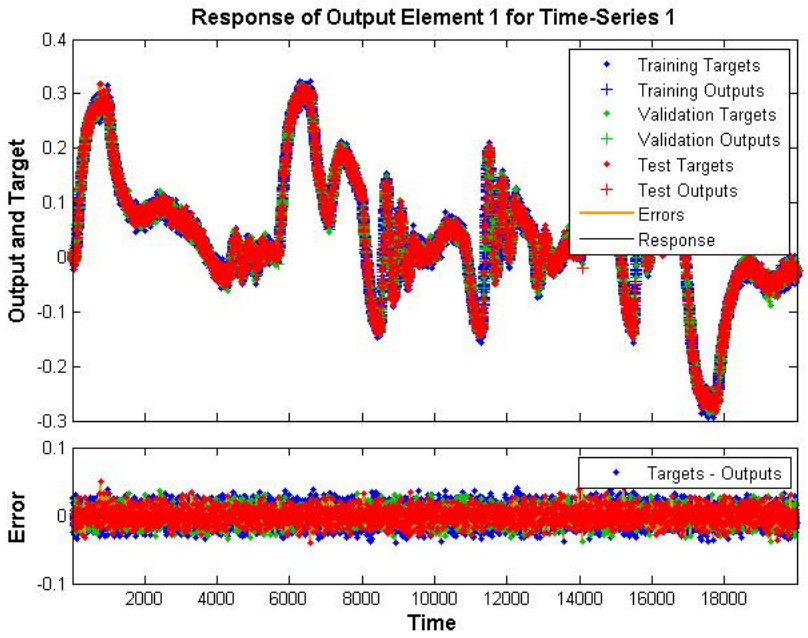

Figure 22. Diagnostic results for surge element of complex NARX

To verify this model, and that the data used to train it was sufficient, the NARX was tested using simple coupled motion between surge and sway since the data used to train this NN included similar coupled motion. The results of plugging the resulting NARX model into the simulator are presented in Figure 23.

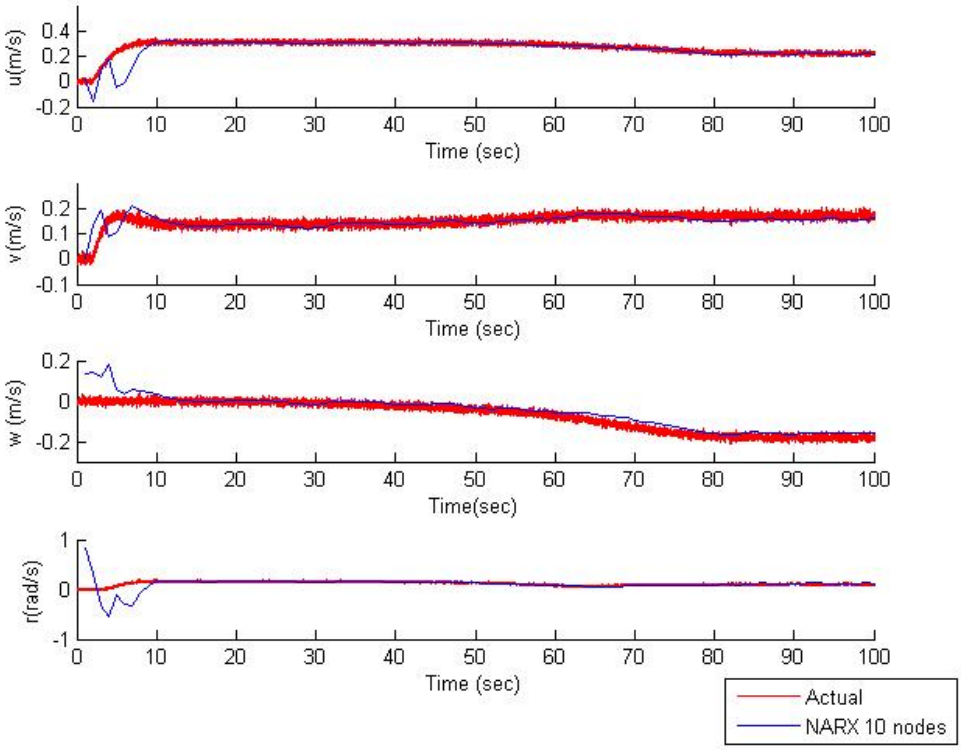

Figure 23. Comparison of surge velocity response

The results presented in Figure 23 show the power of using information rich training data and choosing the right network size. Because a complex and information rich training scheme is used to train the network, when a different input signal is applied the NARX is able to adapt and correctly model the system (with some difficulty in the transient section due to the network response time). This serves to further highlight the importance of the data used to train the NN. Also, since a more complex relationship is modeled, a greater number of nodes are required to adequately model the system. .

2. Neural Network applied to SeaBotix vLBV300

Now that the NN concept, specifically using the NARX architecture, has been validated using the simulator, it is applied to the SeaBotix vLBV300. As discussed in Section V.B, while limited computing power restricts the number of nodes that can be employed, the best results are produced ensuring the richest possible data is used to train the networks.. The NARX experiment applied to the vLBV300 was a stair-step input the

surge direction where only the thrust generated in the surge direction was supplied as input and only *u*, the surge velocity was used as the targeted output. A three node network, using two time delays was created in MATLAB with the error analysis presented in Figure 24. A single node, single delay network was also created but the diagnostic results are not presented as they are very similar to the three node network.

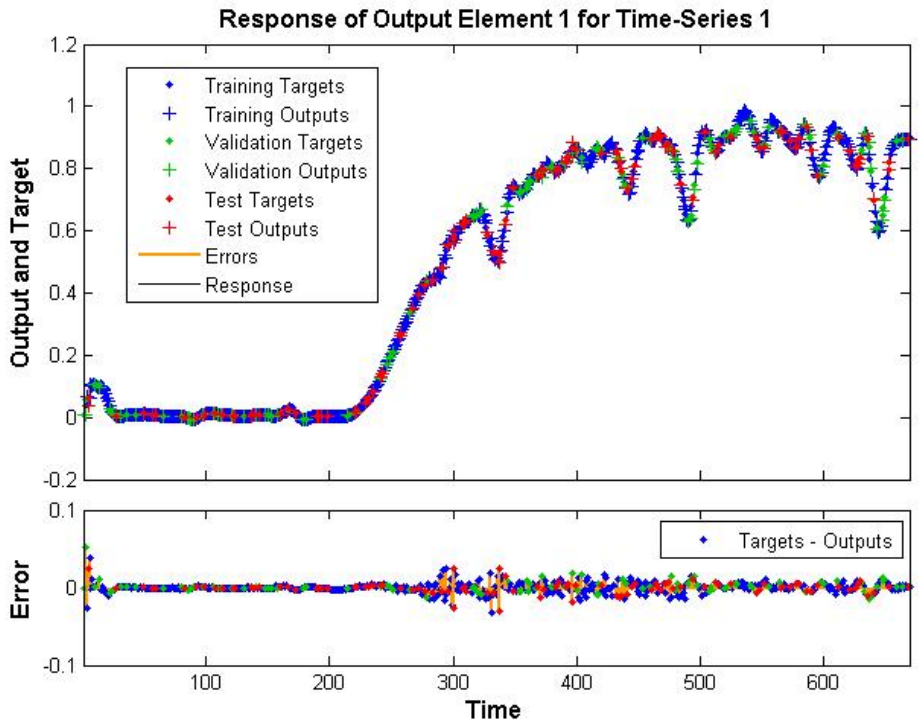

Figure 24. Diagnostic results for velocity mapping, three node, two delay NARX in surge direction only

Again, the error plot demonstrates just how well a very small NN can be trained to follow a non-linear dynamic system. As before, the network was then verified by comparing its outputs to measured data as presented in Figure 25.

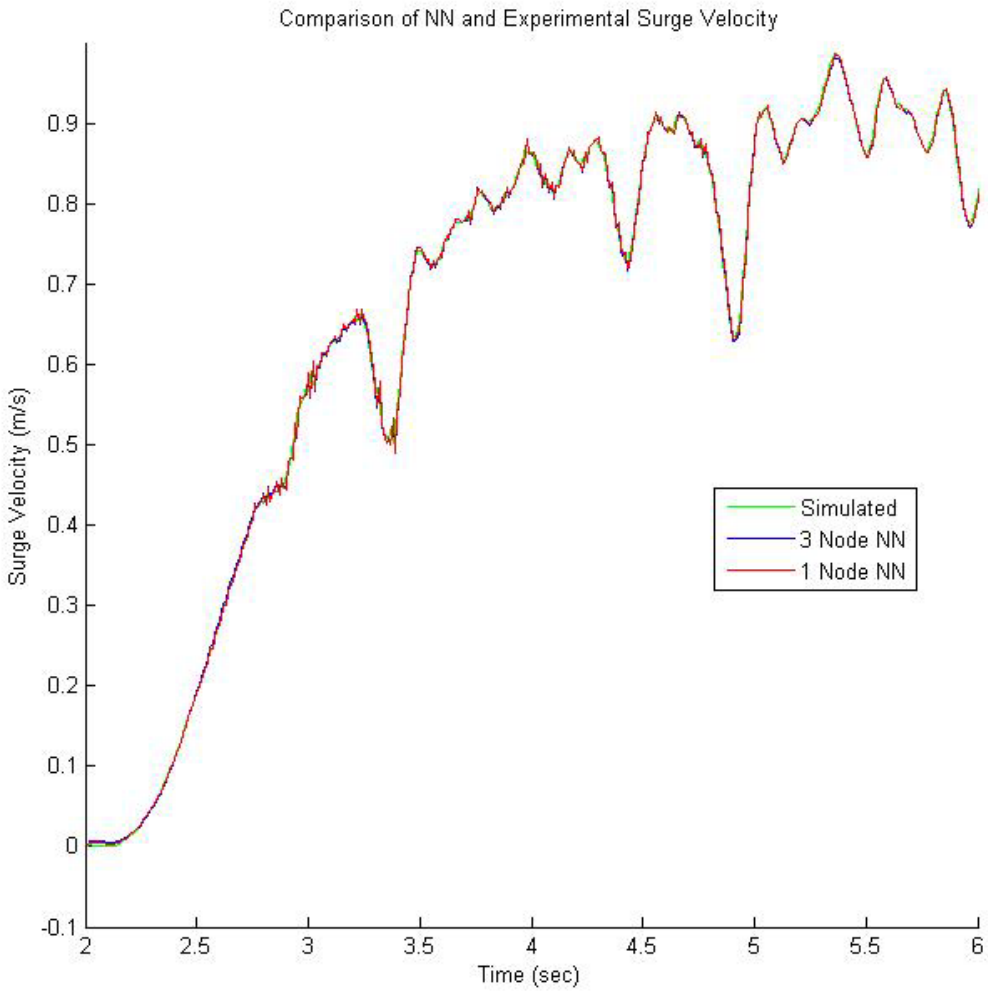

Figure 25. Comparison of three and one node NARXs to measured data

Even the single node NARX was able to match the transient—almost too well: the single node network is able to match the noise inherent to the Vicon system.

Next, a complex run by the SeaBotix coupling the surge, sway, heave, and yaw degrees of freedom was used to train a complex NN. Because of the small size of the tank and a limitation with capturing joystick to PWM commands, the network could not be sufficiently trained to model the system. Based on the simulation results, this is due to not enough data being collected for the various DOF. The simulation results are achieved by using 100 simulated seconds of data with a sample rate of 10 Hz. This yielded over 10,000 data points. It was not possible to collect this much information, or as rich as

information using the actual SeaBotix due to the current experimental setup and reliance on externally collected motion data.

Conceptually, a single well-trained NN will be able to approximate motion in any DOF or number of coupled, controllable DOFs, but as shown that will require a network that is trained, and re-trained, with rich and comprehensive input data. This ideal network may require a large number of nodes depending in the complexity of the system being modeled, but as shown too many nodes will actually degrade the performance of the NN estimator. This training data must contain information on all operating modes, controllable DOF, spanning the full range of motions of the anticipated operations in order to ensure success. This highly accurate model will be useful for both motion prediction and fault detection and assessment. For example, if the vehicle suspects it is damaged due to onboard diagnostics or because of divergence of true motion from another model (perhaps supplied by the RLLS method) it could enter a sub-routine where it performs motion similar to those used to create the simplified NN. Significant deviation from this simplified motion would not only imply damage or obstruction but would also simplify the fault diagnosis process since fewer numbers of thrusters are used as inputs to these networks.

C. GRADIENT ESTIMATOR

The GE is greatly limited by the presence of un-modeled disturbances and measurement noise, and is very sensitive to the values used to tune the filter – particularly step size, p_0 as defined in Section IV.A.3. As in Chapter IV, since both of these are expected to be present in this work the GE is not a good choice for online parameter estimation of a ROV and is not further investigated. This is confirmed by attempting to use the GE to estimate the surge parameters of the Seamus ROV presented in [5]. Recall that the only guarantee of convergence for a GE is sufficient PE, therefore a chirp signal was used to excite the simulated system since it a signal of the highest order of PE and is rich in frequency content as well.

First, the system is excited using a chirp signal (a sinusoid based signal of high PE that increases its frequency over a set period time) with no simulated noise and the estimator step size, p_0, is varied from one to 1000. The results presented in Figure 26 are typical for all p_0 values.

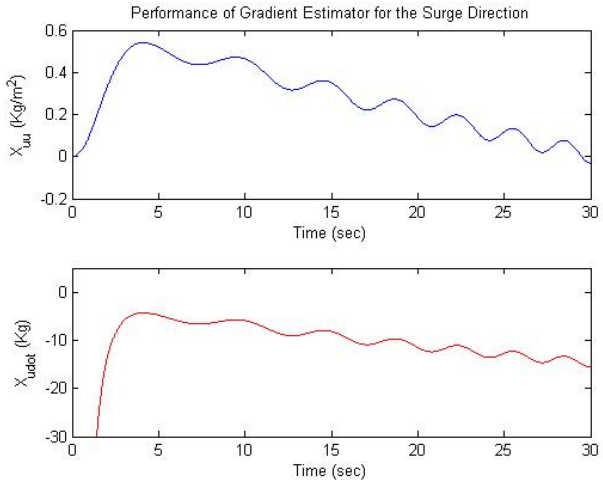

Figure 26. GE results in the surge direction with no noise added

The results, while oscillating, can be seen to be drifting down towards the true values of 61.117 kg/m^2 and 27.08 kg for $X_{u|u|}$ and $X_{\dot{u}}$, respectively. $X_{\dot{u}}$ even arrives in the general vicinity of its true value and then proceeds to oscillate around it. $X_{u|u|}$, however, is very far from the true value even after a 30 second period of excitation. In order for the $X_{u|u|}$ value to arrive at its true value the experiment would have to run for a very long period of time (on the order of thousands of seconds in this example) Because of the small size of the NPS tank long experimental runs are not possible. As such a longer simulated experiment was not conducted since an operational equivalent is not currently possible and because the RLLS estimator is shown to function better in the presence of both noise and uncertainty. Combining this long time to convergence, with the oscillation caused by the choice of p_0, reveals the GE to be a poor choice for this work.

Adding even very low power noise (a magnitude of 0.0001) to the regressor inputs to the filter illustrates the vulnerability of the GE to noise. The results presented in Figure 27 further show why the GE is a poor choice for this work. Because of this, the GE method was not applied to the SeaBotix platform.

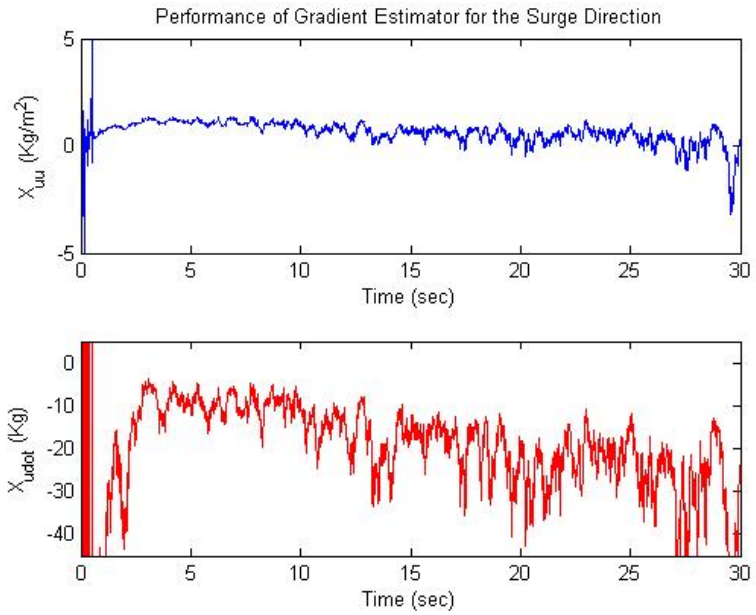

Figure 27. GE results in the surge direction with added noise.

D. BAYESIAN FILTERING

By choosing a zero mean, normal distribution for the probability functions related to the coefficients to be estimated as well as the measurement and process noise distributions the Bayesian filter becomes an extended Kalman filter. In order to successfully apply the Kalman filter equations a system must be observable as derived by Ogata [18].

To check for system observability, first the system must be linearized and written in the canonical state-space form

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{133}$$

The surge direction is considered first by re-arranging Equation (99) into the form of Equation (133) to give

$$
\begin{aligned}
x &= \begin{bmatrix} u \\ a_1 \\ a_2 \end{bmatrix} \\
\dot{x} &= \begin{bmatrix} \dot{u} \\ \dot{a}_1 \\ \dot{a}_2 \end{bmatrix} = \begin{bmatrix} a_{1,i} u_i |u_i| + a_{2,i} X_{prop,i} \\ 0 \\ 0 \end{bmatrix} \\
A &= \begin{bmatrix} 2\sqrt{u_i} a_{1,i} & u_i |u_i| & X_{prop,i} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\
C &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \\
y &= u
\end{aligned}
\tag{134}
$$

In Equation (134) $a_1 = \dfrac{X_{u|u|}}{(m - X_{\dot{u}})}$ and $a_2 = \dfrac{1}{(m - X_{\dot{u}})}$. It is important to note that in Equation (134) *u* refers to the surge velocity in the body frame of reference and in Equation (133) it refers to a control input. Matrices *B* and *D* are ignored here since they do not enter into the observability analysis. Then, from [18], observability of the system is given by

$$
rank \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{pmatrix} = m \tag{135}
$$

where *n* is the size of the state space, or three, in this example. The observability matrix from Equation (135) is then calculated as

$$
\begin{bmatrix} C \\ CA \\ CA^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 2\sqrt{u_i} a_{1,i} & u_i |u_i| & X_{prop,i} \\ (2\sqrt{u_i} a_{1,i})^2 & 2\sqrt{u_i} a_{1,i} u_i |u_i| & \sqrt{u_i} a_{1,i} X_{prop,i} \end{bmatrix} \tag{136}
$$

The determinant of matrix *A* is zero which means that this observability matrix is rank deficient. Therefore, the system is not observable. Since the system is not observable the

extended Kalman filter cannot be applied to this work. This is not an atypical result. While most physical systems are in fact observable, the way they are expressed mathematically may not be [18]. Therefore, the Bayesian approach is not investigated further for this work and it is not applied to the SeaBotix vLBV300.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSIONS

A. SUMMARY

Estimating the hydrodynamic coefficients of an underwater vehicle is a difficult process. The motion of a body in areal fluid is a highly complex, highly coupled, and non-linear process. A propulsion model is developed that maps individual thruster low level PWM commands to generated thrust and a geometric model that converts thrust into body forces and moments. Then, starting with the first principles for the equations of motion for a six DOF submerged vehicle as presented by Fossen [1], several assumptions are introduced to simplify this model and create a parametric representation of the SeaBotix vLBV300 THAUS.

This thesis is focused on applying various system identification techniques to learn the model parameters for the THAUS. Since these system identification techniques are developed for static regression models, a method of converting a dynamic system into an equivalent static system is presented before four system identification techniques are applied: recursive linear least squares (RLLS), computational neural networks (NN), gradient estimator (GE), and a Bayesian filtering method. The GE and Bayesian estimator approaches are not suitable for this work or not applicable and as such no results are compared to the other parametric model based system identification method – the RLLS.

The RLLS approach is applied to learn the coefficients at a low- and high-speed trim condition, respectively. By sequentially exciting the system in individual, decoupled directions, the model parameters can be estimated. Since the vehicle cannot be controlled in the pitch direction, coupled motion between the surge and heave directions are required for system identification. It is shown that these parameters can indeed be estimated online. This is demonstrated on a simulator with known parameters as well as the vLBV300 platform, utilizing an external motion capture system and performing tests in the NPS CAVR test tank. Model accuracy is good in most directions; however the

experimental setup limited the lengths of the data sets, which was particularly constraining for higher-speed data collection as well as angular motion.

Comparing predicted and measured responses, the vehicle response close to the trim condition is accurately captured, but behavior deviates farther away due to the complex dynamics and inherent parameter dependence on vehicle state. Dimensionless parameters are introduced to account for parameter deviation from the trim condition. These dimensionless parameters are learned with their dimensional counterparts. The dimensionless parameter model formulation yielded more accurate predictive capability since the model can correct for slight variations in vehicle velocity (around the trim condition). However, this technique is not applicable to large deviations. To this end, a Reynolds number analysis shows that the low-speed trim condition falls within the laminar-to-transition flow regime, for which linear damping may be a more appropriate model (quadratic damping is assumed, as motivated by vortex shedding effects for turbulent flow). This revised model is investigated for the low-speed trim condition, but without significant improvement in model performance. As a result, it is concluded that operation in this flow transition regime will likely require inclusion of both terms. The original model (with quadratic damping) appears to be appropriate for the higher-speed operations.

A non-parametric approach is also investigated based on a computational NN framework. A simple network is created for both the simulator and the SeaBotix vLBV300 that successfully mapped the thruster force in the surge direction to the surge velocity u. Then a more complicated network is created to map the four controllable DOF thruster forces to their respective body velocities and angular rate. This is accomplished for both the simulator, but the experimental setup did not allow sufficient data to be captured for the physical system. The potential of the NN framework and its ability to capture complex input-output relations inherent in modeling underwater vehicles is demonstrated. The richness of the data used to train a NN is very important. Much like the concept of PE, a NN is only as good as the data with which it is trained. Then, the importance of using a sufficient number of nodes and time delays is demonstrated, but the pitfall of over-fitting is also high-lighted. Because this approach is

non-parametric, the resulting model does not have the same physical meaning as the parametric approaches presented so a direct comparison of the identified parameters is not possible.

B. FUTURE WORK

Based on the results achieved in this work, as well as the limitations identified, several areas for future work have been identified. First, the propulsion model is based on sending low-level commands directly to each thruster. This was beneficial in that it allowed for control of the vehicle in an additional DOF, the roll direction. However, it is a limitation given the small size of the NPS dive tank. Mapping joystick commands directly to the generated thrust will allow for mid-experiment course corrections to avoid collision or further excite the vehicle without throwing off the force input to the filters.

The assumption that the tether dynamics does not affect the vehicle dynamics or pose is observed to be marginal at best during the testing, particularly during operations in the test tank. Further work is required to model the effects of the tether to ensure the more accurate control is possible.

CAVR is acquiring an INS for the SeaBotix vLBV300 in the near future. This will overcome many of the experimental setup limitations encountered in this work. The main advantage is that the sensor will allow a greater range of motions to be tracked, as well as ocean-based operations. Open ocean trials will allow for much longer data runs allowing the filters to have more, and thanks to wave action, richer data to use for convergence. Once this has occurred, the approach and filters developed in this work can be modified to use the measurements provided by the INS in order to improve the accuracy of the filter.

A key assumption that allowed the use of the classical RLLS filter is that the parameters being estimated were not time varying. Because of this, the RLLS is formulated to consider information contained in the entire experiment (i.e., no old data is discarded). Because the hydrodynamic coefficients were shown to vary with vehicle speed and were also shown to be non-scalable to that speed difference, a time-varying approach may prove beneficial and should be investigated. This would not require

substantial further work as the classical RLLS employed in this work can be readily adapted with a “forgetting” factor that allows it to only consider data over a certain period of time. Other approaches for time-varying parameters can be investigated.

Considering the Bayesian filtering approach: while the system as written is not observable and the extended Kalman filter approach is not usable, alternative mathematical formulation of the problem may produce the necessary condition for observability. This will allow for an investigation of not only the extended Kalman filter approach but for other statistical methods based on Bayes’ rule, such as the unscented Kalman filter or a particle filter, which would also prove useful when considering the time-varying parameter problem.

APPENDIX A. 6 DOF MOTION SIMULATOR MATLB CODE

```
function [ydot,etadot,J,out] = fcn(y,eta,tau)
%y=[u;v;w;p;q;r];
%tau=[X;Y;Z;K;M;N];
%eta=[x;y;z;phi;theta;psi];

g = 9.81; % Acceleration due to gravity in meters/second^2 (m/s^2)
m = 20.9; % Mass in kilgrams (kg)
W = m*g; % Weight in Newtons (N)
B = W; % Measured Vehicle Buoyancy (N)

% Moments of Inertia WRT Origin at Half-Length
I_xx = .5587; % kg*m^2
I_yx = 0;
I_zx = 0;
I_xy = 0;
I_yy = .9531; % kg*m^2
I_yz = 0;
I_zy = 0;
I_xz=0;
I_zz = .9; % kg*m^2
I=[I_xx -I_xy -I_xz; -I_yx I_yy -I_yz; -I_zx -I_zy I_zz];

% Center of Buoyancy WRT Origin at Vehicle Nose
x_cb = 0.00; % x-location (m)
y_cb = 0.00; % y-location (m)
z_cb = 0.00; % z-location (m)

% Center of Buoyancy WRT Origin at Vehicle Half Length
% halflength=L/2;
x_cb = 0.00;
y_cb = 0.00; % y-location (m)
% z_cb = -5.016E-1; % z-location (m)
z_cb=-.05;
% Center of Gravity WRT Origin at Vehicle Half Length
x_cg = 0; % x-location (m)
y_cg = 0.00; % y-location (m)
z_cg = 0; % z-location (m)

% Non-Linear Force Coefficients
X_uu = -4.56; % Cross-flow Drag (kg/m)
X_udot = 200.4802; % Added Mass (kg)
X_u = -6.8387;
X_vdot = 0;
X_wdot = 0;
X_pdot = 0;
X_qdot = 0;
X_rdot = 0;
X_wq = 0; % Added Mass Cross-term (kg/rad)
X_qq = 0; % Added Mass Cross-term (kg*m/rad)
X_vr = 0; % Added Mass Cross-term (kg/rad)
```

```
X_rr = 0; % Added Mass Cross-term (kg*m/rad)
Y_vv = -55.006;
Y_v=0;% Cross-flow Drag (kg/m)
Y_rr = 0; % Cross-flow Drag (kg*m/rad^2)
Y_uv = 0; % Body Lift Force and Fin Lift (kg/m)
Y_udot = 0;
Y_vdot = -21.584; % Added Mass (kg)
Y_wdot = 0;
Y_pdot = 0;
Y_qdot = 0;
Y_rdot = 0; % Added Mass (kg*m/rad)
Y_ur = 0; % Added Mass Cross-term and Fin Lift (kg/rad)
Y_wp = 0; % Added Mass Cross-term (kg/rad)
Y_pq = 0; % Added Mass Cross-term (kg*m/rad)
Y_uudr = 0; % Fin Lift Force (kg/(m*rad))
Z_ww = -67.8358;
Z_w=0;% Cross-flow Drag (kg/m)
Z_qq = 0; % Cross-flow Drag (kg*m/rad)
Z_uw = 0; % Body Lift Force and Fin Lift (kg/m)
Z_udot = 0;
Z_vdot = 0;
Z_wdot = -21.3775; % Added Mass (kg)
Z_pdot = 0;
Z_qdot = 0; % Added Mass (kg*m/rad)
Z_rdot = 0;
Z_uq = 0; % Added Mass Cross-term and Fin Lift (kg/rad)
Z_vp = 0; % Added Mass Cross-term (kg/rad)
Z_rp = 0; % Added Mass Cross-term (kg/rad)
Z_uuds = 0; % Fin Lift Force (kg/(m*rad))

K_pp = -103.335;
K_p=0;% Rolling Resistance (kg*m^2/rad^2)
K_udot = 0;
K_vdot = 0;
K_wdot = 0;
K_pdot = -23.890; % Added Mass (kg*m^2/rad)
K_qdot = 0;
K_rdot = 0;
M_ww = 0; % Cross-flow Drag (kg)
M_qq = -129.99;
M_q=0;% Cross-flow Drag (kg*m^2/rad^2)
M_uw = 0; % Body and Fin Lift and Munk Moment (kg)
M_udot = 0;
M_vdot = 0;
M_wdot = 0; % Added Mass (kg*m)
M_pdot = 0;
M_qdot = -46.726; % Added Mass (kg*m^2/rad)
M_rdot = 0;
M_uq = 0; % Added Mass Cross-term and Fin Lift (kg*m/rad)
M_vp = 0; % Added Mass Cross-term (kg*m/rad)
M_rp = 0; % Added Mass Cross-term (kg*m^2/rad^2)
M_uuds = 0; % Fin Lift Moment (kg/rad)
N_vv = 0; % Cross-flow Drag (kg)
N_rr = -.8814;
N_r=0;% Cross-flow Drag (kg*m^2/rad^2)
```

```
N_uv = 0; % Body and Fin Lift and Munk Moment (kg)
N_udot = 0;
N_vdot = 0; % Added Mass (kg*m)
N_wdot = 0;
N_pdot = 0;
N_qdot = 0;
N_rdot = -.4202; % Added Mass (kg*m^2/rad)
N_ur = 0; % Added Mass Cross-term and Fin Lift (kg*m/rad)
N_wp = 0; % Added Mass Cross-term (kg*m/rad)
N_pq = 0; % Added Mass Cross-term (kg*m^2/rad^2)
N_uudr = 0; % Fin Lift Moment (kg/rad)
tauf=tau;
%% Mass Matrices
Ma = -1*[X_udot X_vdot X_wdot X_pdot X_qdot X_rdot;...
    Y_udot Y_vdot Y_wdot Y_pdot Y_qdot Y_rdot;...
    Z_udot Z_vdot Z_wdot Z_pdot Z_qdot Z_rdot;...
    K_udot K_vdot K_wdot K_pdot K_qdot K_rdot;...
    M_udot M_vdot M_wdot M_pdot M_qdot M_rdot;...
    N_udot N_vdot N_wdot N_pdot N_qdot N_rdot];
Mrb = [m 0 0 0 m*z_cg -m*y_cg;...
    0 m 0 -m*z_cg 0 m*x_cg;...
    0 0 m m*y_cg -m*x_cg 0;...
    0 -m*z_cg m*y_cg I_xx -I_xy -I_xz;...
    m*z_cg 0 -m*x_cg -I_yx I_yy -I_yz;...
    -m*y_cg m*x_cg 0 -I_zx -I_zy I_zz];
M = Ma+Mrb;

%% Coriolis and Centripetal Matrix Calculation
%y=[u;v;w;p;q;r];
%tau=[X;Y;Z;K;M;N];
%eta=[x;y;z;phi;theta;psi];
u=y(1);
v=y(2);
w=y(3);
p=y(4);
q=y(5);
r=y(6);
a1 = X_udot*u+X_vdot*v+X_wdot*w+X_pdot*p+X_qdot*q+X_rdot*r;
a2 = X_vdot*u+Y_vdot*v+Y_wdot*w+Y_pdot*p+Y_qdot*q+Y_rdot*r;
a3 = X_wdot*u+Y_wdot*v+Z_wdot*w+Z_pdot*p+Z_qdot*q+Z_rdot*r;
b1 = X_pdot*u+Y_pdot*v+Z_pdot*w+K_pdot*p+K_qdot*q+K_rdot*r;
b2 = X_qdot*u+Y_qdot*v+Z_qdot*w+K_qdot*p+M_qdot*q+M_rdot*r;
b3 = X_rdot*u+Y_rdot*v+Z_rdot*w+K_rdot*p+M_rdot*q+N_rdot*r;
Ca = [0  0  0  0  -a3  a2;...
      0  0  0  a3  0  -a1;...
      0  0  0 -a2  a1  0;...
      0 -a3 a2 0  -b3  b2;...
      a3 0 -a1 b3  0  -b1;...
     -a2 a1 0 -b2  b1  0];
%Fossen's Simplified for UUV

rb1 = zeros(3);
rb2 = [-m*(y_cg*q+z_cg*r) m*(y_cg*p+w) m*(z_cg*p-v);...
    m*(x_cg*q-w) -m*(z_cg*r+x_cg*p) m*(z_cg*q+u);...
    m*(x_cg*r+v) m*(y_cg*r-u) -m*(x_cg*p+y_cg*q)];
```

```
rb3 = [m*(y_cg*q+z_cg*r) -m*(x_cg*q-w) -m*(x_cg*r+v);...
    -m*(y_cg*p+w) m*(z_cg*r+x_cg*p) -m*(y_cg*r-u);...
    -m*(z_cg*p-v) -m*(z_cg*q+u) m*(x_cg*p+y_cg*q)];
rb4 = [0 -I_yz*q-I_xz*p+I_zz*r I_yz*r+I_xy*p-I_yy*q;...
    I_yz*q+I_xz*p-I_zz*r 0 -I_xz*r-I_xy*q+I_xx*p;...
    -I_yz*r-I_xy*p+I_yy*q I_xz*r+I_xy*q-I_xx*p 0];
Crb = [rb1 rb3; rb2 rb4];


C = Ca+Crb;
%% Damping Matrix
d1 = [X_u 0 0 0 0 0;...
    0 Y_v 0 0 0 0;...
    0 0 Z_w 0 0 0;...
    0 0 0 K_p 0 0;...
    0 0 0 0 M_q 0;...
    0 0 0 0 0 N_r];
d2 = [X_uu*abs(u) 0 0 0 0 0;...
    0 Y_vv*abs(v) 0 0 0 Y_rr*abs(r);...
    0 0 Z_ww*abs(w) 0 Z_qq*abs(q) 0;...
    0 0 0 K_pp*abs(p) 0 0;...
    0 0 M_ww*abs(w) 0 M_qq*abs(q) 0;...
    0 N_vv*abs(v) 0 0 0 N_rr*abs(r)];
D =-d1;
out=[D(1,1), d1(1,1),d2(1,1),u];

%% Restoring Force Matrix
%eta=[x;y;z;phi;theta;psi];
phi=eta(4);
theta=eta(5);
psi=eta(6);
gn = [(W-B)*sin(theta);...
    -(W-B)*cos(theta)*sin(phi);...
    -(W-B)*cos(theta)*cos(phi);...
    -(y_cg*W-y_cb*B)*cos(theta)*cos(phi)+(z_cg*W-
z_cb*B)*cos(theta)*sin(phi);...
    (z_cg*W-z_cb*B)*sin(theta)+(x_cg*W-x_cb*B)*cos(theta)*cos(phi);...
    -(x_cg*W-x_cb*B)*cos(theta)*sin(phi)-(y_cg*W-y_cb*B)*sin(theta)];
gnd=gn;

%% Y dot calculation
 ydot = M\(tauf-C*y-D*y-gn);
J1=[cos(psi)*cos(theta) -sin(psi)*cos(phi)+cos(psi)*sin(theta)*sin(phi)
sin(psi)*sin(phi)+cos(psi)*cos(phi)*sin(theta);...
    sin(psi)*cos(theta) cos(psi)*cos(phi)+sin(phi)*sin(theta)*sin(psi)
-cos(psi)*sin(phi)+sin(theta)*sin(psi)*cos(phi);...
    -sin(theta) cos(theta)*sin(phi) cos(theta)*cos(phi)];
J2= [1 sin(phi)*tan(theta) cos(phi)*tan(theta);...
    0 cos(phi) -sin(phi);...
    0 sin(phi)/cos(theta) cos(phi)/cos(theta)];
J=[J1, zeros(3); zeros(3),J2];

etadot = J*y;
```

APPENDIX B. SIMULINK DIAGRAMS

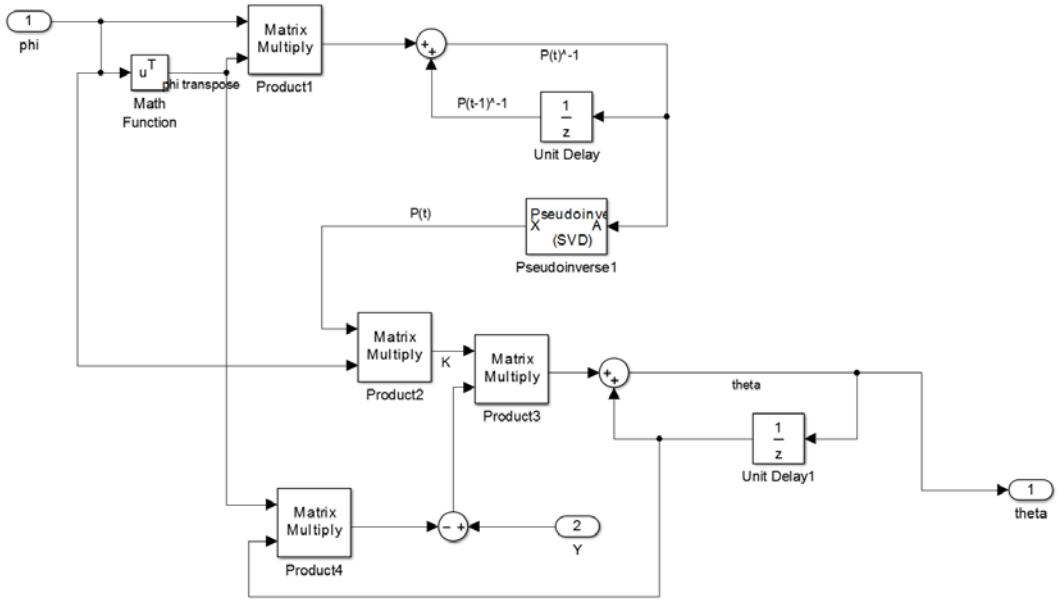

Figure 28. RLLS estimator

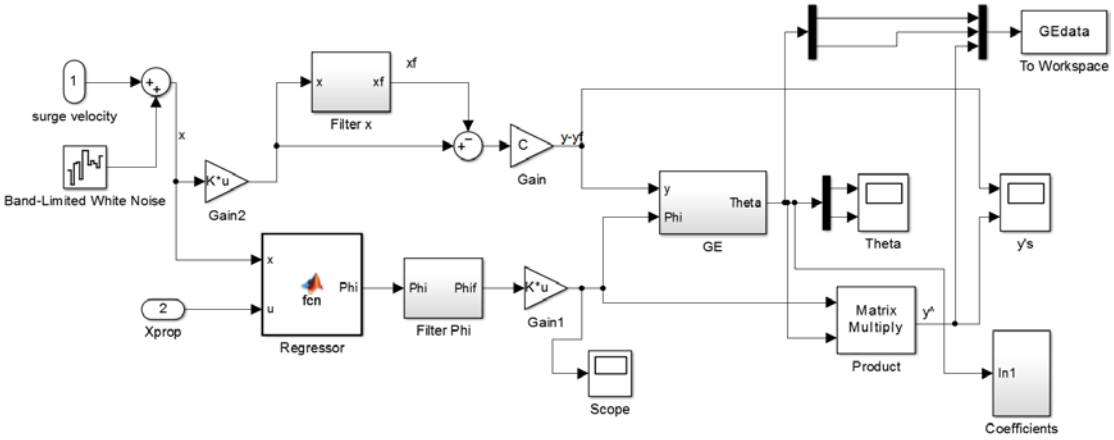

Figure 29. Gradient estimator

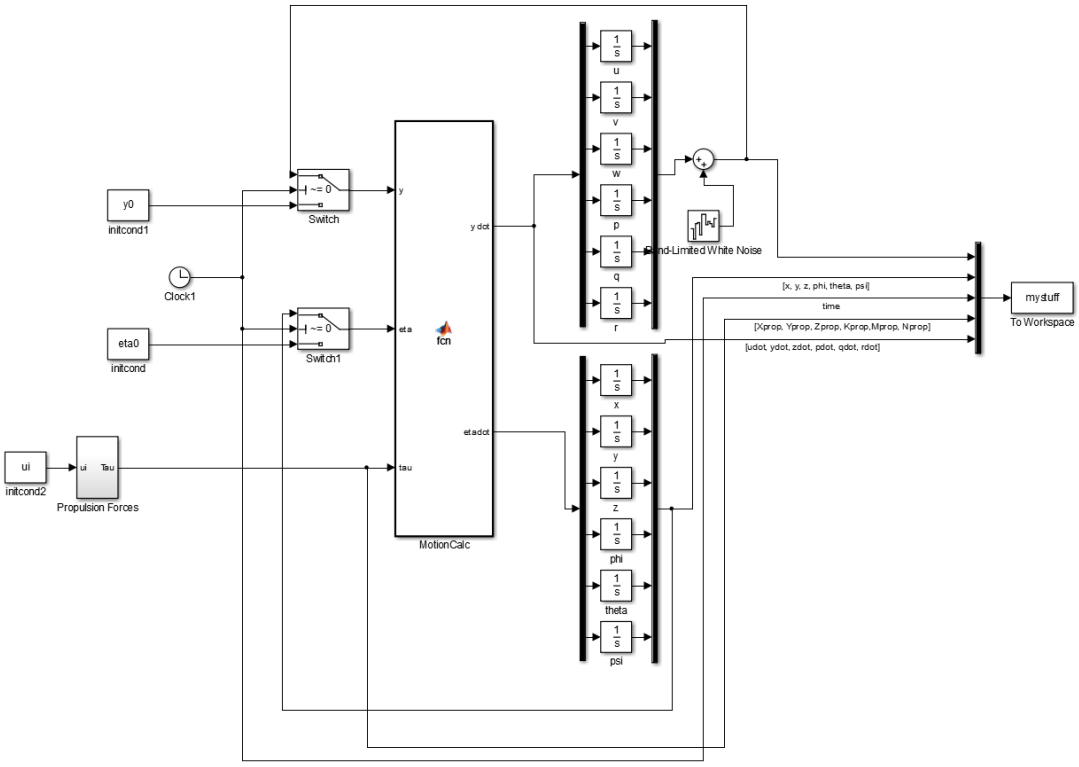

Figure 30. Simulator

LIST OF REFERENCES

[1] T. Fossen, *Guidance and Control of Ocean Vehicles.* New York: John Wiley & Sons, 1994.

[2] J. J. E. Slotine and W. Li, "On-line parameter estimation," in *Applied Linear Control*. Englewood, NJ: Prentice-Hall, Inc, 1991, ch. 8, sec. 7, pp. 358–381.

[3] S. Thrun, W. Bulgard, and D. Fox, "Probabilistic robotics." Cambridge, MA: MIT Press, 2005.

[4] S. Sastry and M. Bodson, "Adaptive control: Stability, convergence, and robustness,=." Englewood Cliffs, NJ: Prentice Hall, 1989.

[5] H. H. Chen *et al.*, "Identification of hydrodynamic properties for a remotely operated vehicle using projective mapping method." Kaohsiung, Taiwan: Institute of Undersea Technology and Asian Pacific Ocean Research Center.

[6] S. M. Doherty, "Cross body thruster control and modeling of a body of revolution autonomous underwater vehicle," M.S. thesis, Dept. Mech. Eng, Naval Postgraduate School, Monterey, CA, 2011.

[7] T. Prestero, "Verification of a six-degree of freedom simulation model for the REMUS autnonmous underwater vehicle," M.S. thesis, Dept. Mech. Eng, Dept. Ocean Eng.," MIT, MA, 2001.

[8] K. Astrom and B. Wittenmark, "Real-time parameter estimation," in *Adaptive Control*. Boston, MA: Addison-Wesley, 1994, ch 2, sec. 2, pp. 42–52.

[9] Y. H. Eng, *et al.*, "Estimation of the hydrodynamics coefficient of an ROV using free decay pendulum motion," *Engineering Letters,* vol. 16, no. 3, Aug 2008.

[10] F. Scarselli and A. Tsoi, "Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results," *Neural Networks,* vol. 11, no. 1, pp. 15–37, Jan. 1998.

[11] N. du Toit and J. Weiss, "Real-time dynamic model learning and adaptation for underwater vehicles," submitted for publication.

[12] C. Andrieu *et al.*, "On-line parameter estimation in general state-space models," in *Decision and Control 2005 European Control Conference*, Seville, Spain, 2005. DOI: 10.1109/CDC.2005.1582177.

[13] P. Kugler, "Online parameter identification in time-dependent differential equations as a non-linear inverse problem," *European Journal of Applied Mathematics*, vol. 19, pp. 479–506. DOI 10.1017/S0956792508007547.

[14] W. Juan *et al.*, "Research of system identification method for underwater vehicle based on neural network," in *Proceedings of the 2007 International Conference on Wavelet Analysis and Pattern Recognition*, Beijing, China. 2007.

[15] SeaBotix, Inc. "vLBV300 MiniROV systems." [Online]. Available: http://www.seabotix.com/products/pdf_files/vLBV300.pdf

[16] R. P. Lippmann, "An introduction to computing with neural nets," *ASSP Magazine, IEEE*, vol. 4, no. 2, pp. 4–22, Apr. 1987. DOI 10.1109/MASSP.1987.1165576.

[17] Y. Hao and B.M. Wilamowski, "Levenberg-Marquardt training," *Industrial Electronics Handbook*, vol. 5, chap. 12, pp. 12-1 to 12-15, Boca Raton, FL: CRC Press, 2011.

[18] K. Ogata, "Analysis of Control Systems in State Space," in *Modern Control Engineering*. Englewood Cliffs, NJ: Prentice-Hall, 2002, ch. 11, sec. 7, pp. 786-790.

[19] F. G. Bahrke, "On-Line Identification of the Speed, Steering and Diving Response Parameters of an Autonomous Underwater Vehicle from Experimental Data," M.S. thesis, Dept. Mech. Eng, Naval Postgraduate School, Monterey, CA, 1992.

[20] F. M. White, *Fluid mechanics*. Singapore: McGraw-Hill, 1999

[21] E. Lavresky, "System identification," class notes for CDS 270-1, Dept. of Control and Dynamical Sys., California Institute of Technology, 2006.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California